AD NUMBER
AD914190
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; 02 NOV 1973. Other requests shall be referred to Air Force Avionics Laboratory, AFAL/AAA, Wright-Patterson AFB, OH 45433.**
AUTHORITY
afal ltr dtd 7 oct 1977

AFAL-TR-73-297

MULTILATERATION SOFTWARE DEVELOPMENT (PHASE II)

J. Weinberg, Principal Investigator
M. Harris, E. Knobbe, E. Kopitzke, G. Kochmann
(Northrop Corporation)

E. Martin, V. Calbi, M. Bittner
(Magnavox Research Laboratories)

TECHNICAL REPORT AFAL-TR-73-297

September 1973

"TEST + EVALUATION"
Distribution limited to U.S. Government agencies only; other requests for this document must be referred to the Air Force Avionics Laboratory (AFAL/AAA), Wright-Patterson Air Force Base, Ohio 45433.

Air Force Avionics Laboratory
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio

Copy No.

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

MULTILATERATION SOFTWARE DEVELOPMENT

(PHASE II)

J. Weinberg, M. Harris, E. Knobbe, E. Kopitzke, G. Kochmann
(Northrop Corporation)

E. Martin, V. Calbi, M. Bittner
(Magnavox Research Laboratories)

"TEST + EVALUATION"

Distribution limited to U.S. Government agencies only; other requests for this document must be referred to the Air Force Avionics Laboratory (AFAL/AAA), Wright-Patterson Air Force Base, Ohio 45433.

FOREWORD

This report, AFAL TR-73-297, covers work done by the Electronics Division of the Northrop Corporation, Palos Verdes, California, under Contract F33615-72-C-1607 and project 6095 02 04 for the US Air Force Avionics Laboratory, Wright-Patterson AFB, Ohio 45433. The USAF program monitor is J A Biernacki (AFAL/AAA). Research on this report was conducted from May 72 to Jun 73 and the report submitted Jul 73. The Magnavox Research Laboratory, Torrance, California, participated in this effort under subcontract to the Northrop Corporation. Contributors to this report are J Weinberg, M Harris, E Knobbe, E Kopitzke, G Kochmann of the Northrop Corporation and E Martin, V Calbi, M Bittner of the Magnavox Corporation.

The authors hereby gratefully acknowledge the many contributions -- technical, administrative, and philosophical -- made to this development program by Mr John Biernacki, who has served as the AFAL Project Engineer from program inception to date.

This technical report has been reviewed and is approved.

Cozier S. Kline

COZIER S KLINE, Colonel, USAF
Chief, System Avionics Division
AF Avionics Laboratory

ABSTRACT

Historically, vehicle-borne, radio-hybrid navigation system software has too often been designed around preselected navigation hardware on an ad hoc, system-by-system basis. In these developments, little attention has been paid to the inherent physical and functional commonality which underlies much of this superficially quite different software. This report documents the methods, and the very promising results, of the second phase of a software development effort directed at identifying and specifying a standardized, modular, flexible, radio-hybrid navigation system software processor.

The machine-and-language-independent (MLI) processor specification which has in particular been developed to date has been designed so that -- with appropriate, minor, system-specific tailoring -- it can serve as the basic specification for the navigation software development for any specific system, within a wide range of navigation hardware equipment configurations and mission requirements. These currently include any combination of radio (LOS or earth mode), inertial, AHRS, and CADS navigation equipments, as well as the requirements associated with most military and civil aircraft missions and usages. In addition, the MLI processor has been carefully structured to allow for easy accommodation of additional navigation hardware processing requirements.

The second-phase effort used as its point of departure and developmental framework the basic guidelines, concepts and algorithms established in the initial phase. These include, in particular, the exclusive use of vector-matrix algorithm formulations, processor organization into basic, building-block function- and hardware-specific modules and submodules, the use of a single, mission-phase switchable, computational reference frame, and the use of partitioned, modularly organized Kalman filtering techniques. The overall second-phase effort itself consisted of two main, more or less sequential developments: (a) the extension and refinement of the MLI processor capabilities beyond its first-phase level, and (b) the initial development of a specialized, higher-order language navigation program using the MLI processor specification as a basis.

The improvements of the MLI processor accomplished in the second phase included (a) extension of its navigation hardware applicability to allow use of cheaper AHRU/CADS equipment, either in lieu of or as a backup to an IMU; (b) further development and refinement of a novel and promising radio-autonomous navigation technique; (c) extension and refinement of processor and navigation equipment initialization and alignment techniques; (d) development of a completely partitioned and modularized Kalman filter; and (e) development of a complete set of processor mode control and switching logic specifications. In particular, one of the initialization algorithms developed is a new and powerful one which allows undegraded Kalman filter use of radio pseudoranging measurements, despite large LOS directional uncertainties. Further, the Kalman filter partitioned modularity was achieved without artificial (and performance-degrading) decoupling of interpartition system error dynamics.

Time and money constraints permitted development of the specialized FORTRAN IV/IBM 370 processor program only to a very limited stage. Specifically, all the principal navigation modules required for a single assumed LOS/inertial navigation hardware configuration and navigation mode of operation have been programmed and checked out (for fixed inputs only); no mode switching or control modules have been programmed. However, even this limited level of development was intended (and has served)to accomplish two purposes. First, it provided a learn-by-doing vehicle for the broadly experienced programmer assigned the task, to assay the viability of the MLI processor specifications from the standpoint of real-time programming in either an HOL or a machine-specific language. The preliminary conclusions reached in this regard are that the MLI specification provides the programmer with an extremely flexible and easily modifiable, but standardized approach to programming real-time, multisensor, Kalman (or non-Kalman) navigation system software for any airborne digital computer. In addition, it places overall program efficiency and balance (with regard to execution time and memory requirements) much more completely under the control of the programmer than traditional types of specification.

The second purpose accomplished lies in the fact that the programmed modules thus far developed constitute a nucleus-in-being for further development of either a processor evaluation simulation program on the one hand, or a standardized, real-time HOL master navigation program for subsequent machine-specific translation, on the other.

CONTENTS

CONTENTS(Continued)

CONTENTS(Continued)

ILLUSTRATIONS

TABLES

TABLES (Continued)

TABLES (Continued)

SYMBOL GLOSSARY

1. COORDINATE REFERENCE FRAME
(all frames Cartesian)

Symbol	Unit Vectors	Frame Definition
I	$\vec{I}_1, \vec{I}_2, \vec{I}_3$	Earth centered, non-rotating w/r to fixed stars. $\vec{I}_1$ = earth's north polar axis direction, $\vec{I}_2$ = normal to $\vec{I}_1$ in direction of point of Aries, $\vec{I}_3 = \vec{I}_1 \times \vec{I}_2$.
E	$\vec{E}_1, \vec{E}_2, \vec{E}_3$	Earth centered, earth-fixed. $\vec{E}_1 = \vec{I}_1$, $\vec{E}_2$ = normal to $\vec{E}$ in plane of Gnwch Meridian, $\vec{E}_3 = \vec{E}_1 \times \vec{E}_2$.
P	$\vec{P}_1, \vec{P}_2, \vec{P}_3$	Platform reference frame. Center at center of rotation of platform (for gimballed or floated platform), or at defined point in platform (for strapdown platform). Orthogonal $\vec{P}_1, \vec{P}_2, \vec{P}_3$ directions fixed in platform
A	$\vec{A}_1, \vec{A}_2, \vec{A}_3$	Airframe frame. Concentric with P frame. Orthogonal $\vec{A}_1, \vec{A}_2, \vec{A}_3$ directions fixed in aircraft.
L	$\vec{L}_1, \vec{L}_2, \vec{L}_3$	Local geographic vertical, wander azimuth frame. Concentric with P frame. $\vec{L}_1$ = up, $\vec{L}_2$ and $\vec{L}_3$ normal to $\vec{L}_1$ and each other.
C	$\vec{C}_1, \vec{C}_2, \vec{C}_3$	Computational (Computer) frame. May be any of the other frames defined here, as specified (C frame subscripting is often omitted for compactness of expression)

1. COORDINATE REFERENCE FRAME (Continued)

Symbol	Unit Vectors	Frame Definition
EF	$\vec{EF}_1, \vec{EF}_2, \vec{EF}_3$	Earth-fixed, but not necessarily earth-centered frame. (e.g. ,local tangent plane frame).
AEM	AEM_1, AEM_2, AEM_3	Emitter air frame; centered at emitter

2. INTERFRAME RELATIONSHIPS

Symbol	Definition
$T_{F2/F1}$	Orthonormal 3x3 matrix transformation from frame F1 to frame F2
$\omega_{F2/F1}$	Vector angular rate of frame F2 w/r to frame F1

3. BASIC NAVIGATION DISPLACEMENT VECTORS

Vector Symbol	Displacement Vector Definition
$P=p_E$	Position vector from center of earth to center of P frame.
E	Position vector from center of earth to center of EM frame.
R	Estimated LOS range vector (= p-e = P-E)
R_m*	Measured LOS range scalar
C	Position vector from center of earth to center of EF frame.
P	Position vector from center of EF frame to center of P frame.
e	Position vector from center of EF frame to center of EM frame.

*scalar

3. BASIC NAVIGATION DISPLACEMENT VECTORS (Continued)

Vector Symbol	Displacement Vector Definition
P_s	Position vector from center of earth to the subaircraft surface ellipsoid point.
E_s	Position vector from center of earth to the subemitter surface ellipsoid point.
A	Position vector from center of earth to center of user antenna.
a	Position vector from center of EF frame to center of user antenna.
d_u	Position vector from center of P frame to center of user antenna.
d_{EM}	Position vector from center of emitter frame to center of emitter antenna
r	Unit LOS range vector
τ	Unit earth mode range vector (user end)
τ_{EM}	Unit earth mode range vector (emitter end)

4. ALTITUDE SCALARS

Symbol	Altitude Definitions
h *	User altitude scalar.
h_{EM}*	Emitter altitude scalar.

*scalars

5. VELOCITY AND ACCELERATION VECTORS

Vector Symbol	Vector Definition
v	Velocity of center of P frame with respect to the earth (i.e., with respect to the E or EF frames)
$v_{EM,\dot{e}}$	Velocity of center of EM frame with respect to the earth
$\omega_{F2/F1}$	Angular rate of frame F2 with respect to frame F1.
f	Specific force acting on aircraft (taken to act at center of P frame)
$G(P)$	Sum of all celestial mass attraction gravitation accelerations acting on aircraft (taken to act at center of P frame) minus same sum acting at center of earth.
$g(P)$	G(P), plus centripetal acceleration at center of P frame due to rotation of the earth.
v_W	Wind velocity vector
v_{AS}	Airspeed velocity vector
β	Maneuver acceleration vector

6. OPERATORS

Operator Symbol	Operator Definition
$\frac{d_F a}{dt}$	Time rate of change of any vector a w/r to frame F (F subscript optional).
$(ax)_F$	Cross-product matrix associated with any vector a in frame F.
$\|a\|$	Length of any vector a.
$\hat{a}, \hat{T}, \hat{s}$	Estimated value of any vector a, of any matrix T, of any scalar s
$\Delta_c a$	Change in a in the basic computational interval Δt.
$\int_{\Delta t} a dt$	Integral of a over the interval t to t + Δt.

6. OPERATORS (Continued)

Operator Symbol	Operator Definition
Σ	Signal addition and/or subtraction
$\int$	Signal integration
δa	Error in computed vector a
$\bar{a},\bar{T},\bar{s}$	Time average of any vector a, of any matrix T, of any scalar s
T	Superscript indicating matrix transposition

7. KALMAN FILTER VECTORS AND MATRICES

See Table XXXIII, page 138.

8. MODULE-SPECIFIC VECTORS, MATRICES, SCALARS

See MLI module operations summary tables (Section III).

SECTION I

INTRODUCTION AND SUMMARY

This document comprises the final report for the second phase of the joint Northrop (prime contractor)/Magnavox (sole subcontractor) software development effort entitled "Navigation by Multilateration to Fixed and Moving Reporting Emitters," Project No. 6095, under Contract No. F33615-72-C-1607. However, since this second phase consisted in large part of a broadening, deepening, systematizing and programming of the basic concepts formulated in the initial phase, much Phase I material -- amended and/or extended as necessary -- has been included to make this report as nearly self-contained as practical.*

The principal results of the overall (two-phase) development are:

a) MLI Processor Specification. The development of a machine and language-independent (MLI) specification of a standardized, modular, flexible, multi-application, radio-hybrid navigation system software processor.

b) HOL (FORTRAN IV/IBM 370) Version. The partial development of a functionally limited, FORTRAN IV/370 programmed version of the above processor.

In particular, the latter, specialized HOL (higher order language) version was developed using the former, generalized (and even higher level language) MLI specification as a basis. Although limited in both function and scope, and incomplete (as an overall navigation processor), the HOL program development nevertheless effectively served its intended purpose as a trial horse for the viability of the MLI specification as a basis for development of navigation software for a specific application.

The report is organized to present both the MLI processor specification and the specialized HOL version in a logically sequential manner, closely paralleling the order and sequence of their actual development. Following this introductory section, Section II presents a generalized description of the processor with regard to its scope and role, its mission and hardware applicability, and its basic structural and operational characteristics, together with the rationale underlying each of aspects and features. With Section II as a background, Section III presents the detailed modular MLI processor specification itself, in terms of standardized, module-by-module operations and input/output summary

*However, since principal emphasis in this report has of course been placed on the Phase II effort in particular, the Phase I Final Report (AFAL-TR-72-80, May 1972) can and should serve as useful background reference material for this document.

tables, and summary logic flow diagrams. Finally, Section IV presents the MLI-based, HOL program processor version, including a programmer-oriented description as well as the actual FORTRAN program listing. Extensive appendices are included at the end of the report as detailed and appropriate backup for the text of these main sections.

Based on the results obtained in the two-phase processor development to date, it is evident that the MLI specification presented in this document, although still incomplete in some regards, already constitutes an extremely powerful tool for the standardized and systematic development of modular, flexible, compact, and efficient navigation system software, for a wide range of specific mission applications and navigation hardware configurations. In fact, recognition of this within both Northrop and Magnavox has already led to its adoption as the basic navigation software approach for several applications, at both the proposal and contractual levels. In particular, it has been adopted by Northrop for its use in the initial, AFAL MRV hardware/software configuration definition phase.

Further, it is also already evident that the concepts and techniques employed to date for development of the navigation-only processor presented here can probably be applied nearly intact, not only to development of additional, wholly compatible software for processing types of navigation sensor not yet specifically considered (e.g., doppler radar, DF and angulation equipment, etc.), but also for other, non-navigation but navigation-related avionics software (e.g., weapon delivery, steering, guidance, etc.). Overall, this suggests the extremely attractive possibility that in the not too distant future, large portions of the overall avionics software packages associated not only with a single weapons system, but with whole classes of weapons systems, may be developable on a standardized, modular, partly interchangeable basis. Significant overall cost savings, both developmental and operational, could thereby be clearly achieved.

Finally, it should also be noted that the FORTRAN IV processor version developed to date constitutes a sizeable nucleus-in-being for a possible, general processor evaluation simulation program. Such a program, especially if used as a central preliminary and/or an adjunct to development of the standardized avionics software mentioned above, would quickly pay for its development cost in terms of avoidance of later pitfalls in checkout of the actual system software, when the design has hardened and is difficult and costly to change.

SECTION II

PROCESSOR DESCRIPTION

This section presents a general description of the processor evolved to date (i.e., as presented in Section III of this report), together with the rationale which underlay its development into this form.

1. FUNCTIONAL SCOPE AND ROLE

Early Phase I effort was mainly concentrated on the essential preliminary of scoping the processor to be developed -- within the time and level-of-effort constraints of that phase -- so as to provide a sound point of departure for subsequent evolution of the processor in later development phases. In this connection, Figure 1 shows the overall airborne avionics equipments data processing requirements for a generic, aircraft weapons delivery system. In this diagram, the block labeled navigation processing plays a central role. This role is further emphasized in Figure 2, which compresses all the required processing into two areas: basic navigation processing, and all other navigation (or navigation-related) processing. This is not an arbitrary, but rather a natural and useful division of the overall avionics processing requirements, as follows.

The meaning of the basic navigation processing intended here is most easily conveyed in terms of the basic outputs -- i.e., user vehicle 3-d position, velocity, attitude, and attitude rate -- it produces. Thus defined, basic navigation processing comprises a central computational role, with respect to which all other computations are peripheral in the sense that they all require one or more of its outputs as inputs, while the converse is not true.

To clarify this, the basic navigation processing computations require some subset of data only from the IMU (acceleration, attitude and, if available, attitude rate), the AHRU (attitude and attitude rate), the transceiver (emitter signal phase and frequency shift, and, when reported, emitter antenna position and velocity), and the CADS (altitude and true airspeed), to continuously generate all the basic navigation outputs, as well as appropriate IMU control and transceiver rate-aiding signals. On the other hand, the inputs to those areas of the overall processing which relate to control of all other onboard avionics hardware equipments can always be simply derived from the basic navigation outputs. For example, in the case of ILS, the required inputs might typically be runway-relative, aircraft horizontal position, horizontal velocity, altitude, vertical velocity, roll, pitch, heading, and turn rate; these can all be simply derived from the basic navigation outputs, if they are expressed in a common earth-fixed reference frame, and if augmented only by runway location data referenced to the same frame.

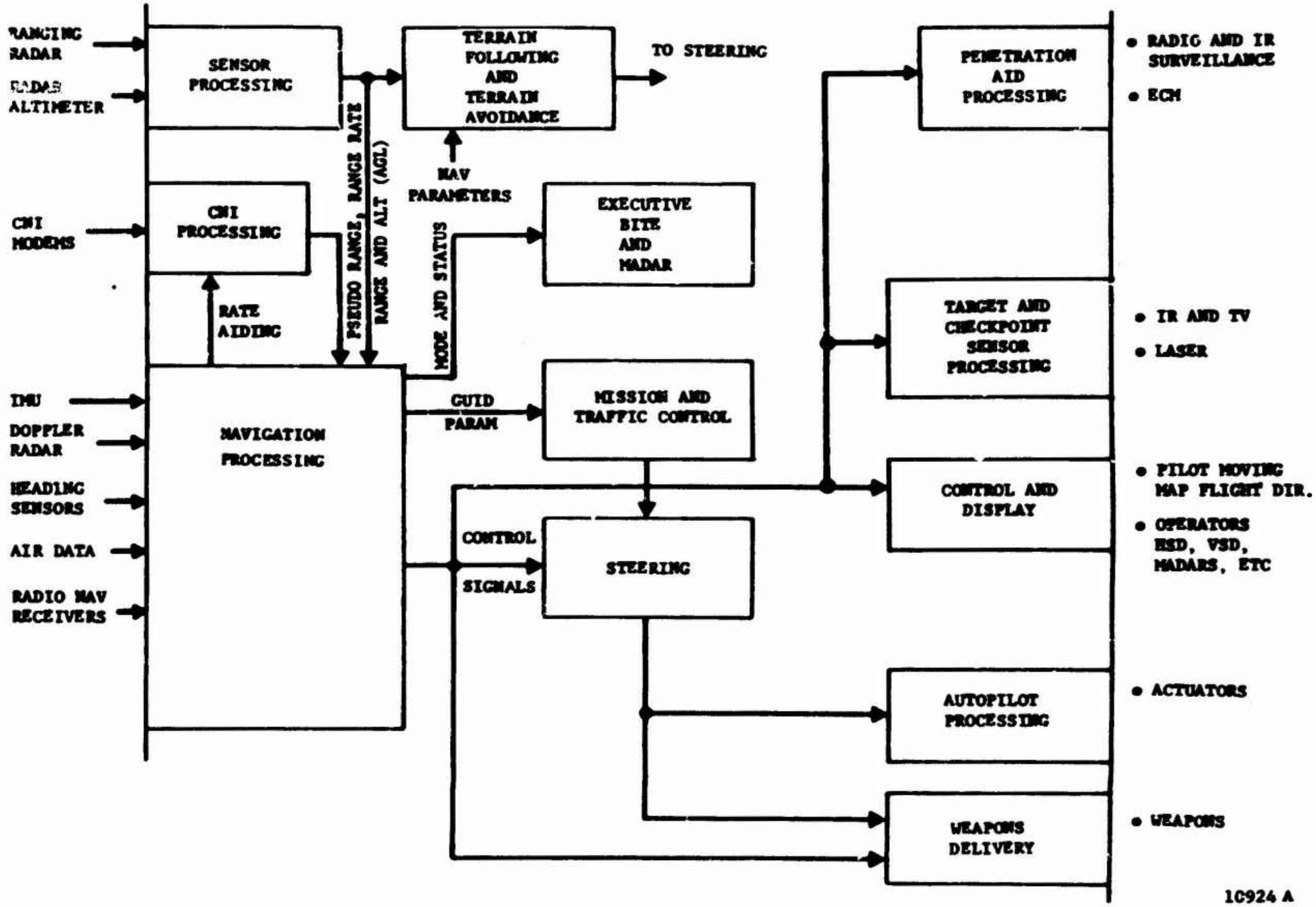
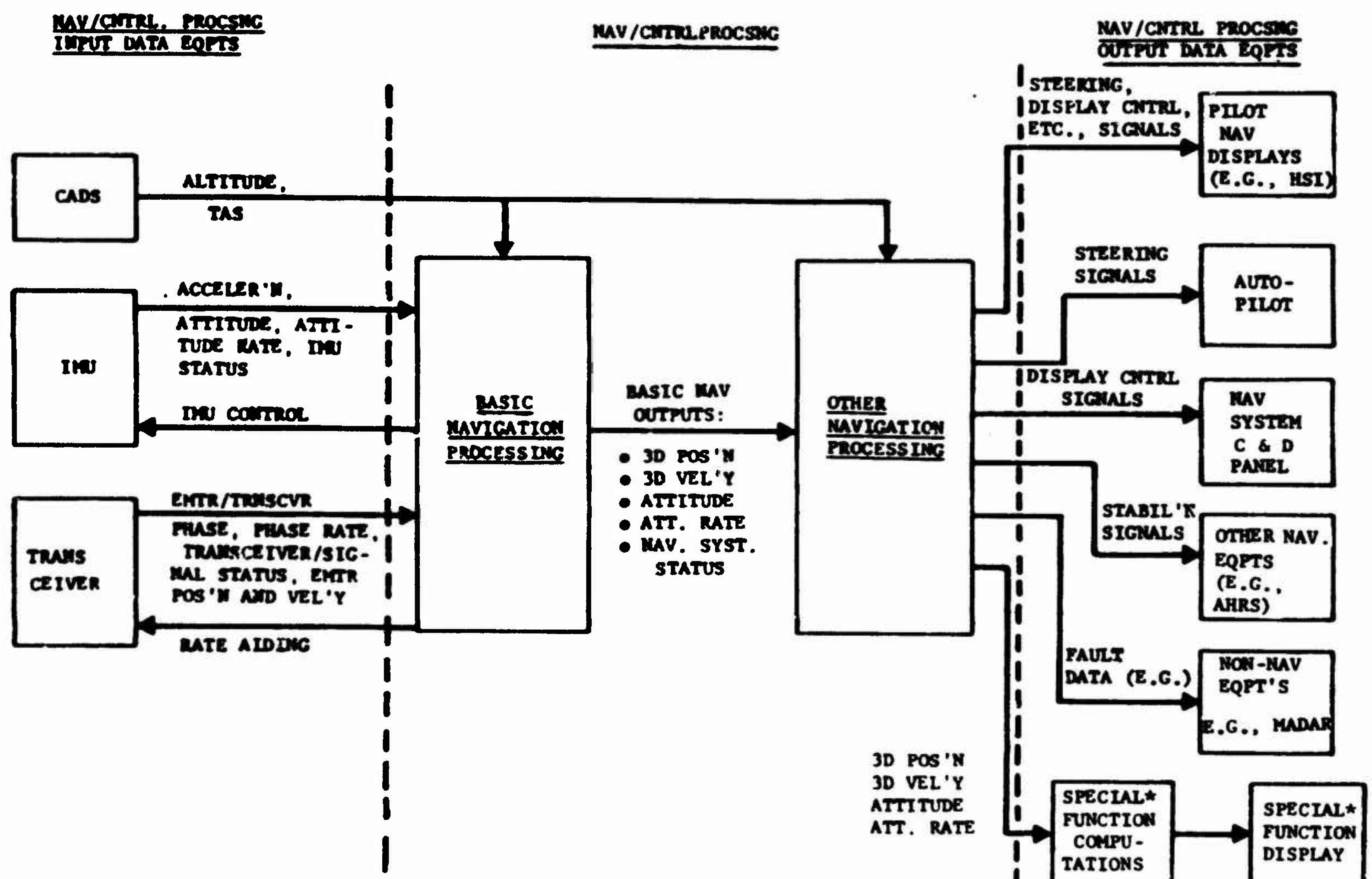


Figure 1. Avionics Data Processing



* E.G., ILS, WEAPON OR CARGO DELIVERY, ETC.

Figure 2. Basic Navigation/Navigation Interface Information Flow

11122A

Phase I processor functional scope was therefore limited to basic navigation processing, thus defined, to enable the attainment of a reasonable first-stage level of processor definition, within the constraints imposed by Phase I level-of-effort and duration.

2. MISSION/HARDWARE APPLICABILITY

Table I summarizes in broad terms the results of a brief Phase I survey of aircraft missions to which the processor under consideration here might apply. Source material for this survey included government and industry sponsored reports, trade periodicals, and consultation with mission requirements specialists at Northrop and Magnavox.

In particular, Table I shows three levels of accuracy requirements -- moderate (about 1 mile DRMS), high (a few hundred feet, DRMS), and very high (under one hundred feet, DRMS) -- for a wide class of military and non-military missions, and during both enroute and objective area phases of the mission. In general, enroute phase accuracy requirements are seen to be uniformly moderate, while objective area requirements range from moderate (e.g., for near-destination checkpoint acquisition in an intercontinental 747 commercial carrier flight) to very high (e.g., for an A-7 close-support mission in zero visibility).

Table I also summarizes in qualitative terms the typical maneuver profiles for the same broad class of missions. Again, there is a marked distinction between the enroute phase, where maneuvers are infrequent and mild (e.g., a long-range patrol mission) and the objective area phase, where they are in many cases frequent and violent (e.g., ground fire evasion maneuvers during a low-altitude, tactical observation mission).

Finally, in Table I, two broad classes of navigation computation coordinate system -- geodetic referenced and locally referenced -- are identified, together with their use by mission and mission phase.

The geodetically referenced coordinate systems, which are most frequently used and useful for long-range, point-to-point navigation, are referenced to earth-centered, earth-fixed frames (e.g., the unit vector triad composed of the unit vector along the earth's polar axis, and two equatorial plane unit vectors, one in the Greenwich Meridian plane, and the other in the 90° Meridian plane). Typical of such coordinate systems in common use are latitude-longitude-altitude (normal and transverse polar), and local vertical direction cosines-altitude.

The second broad class -- the locally referenced coordinate systems -- are most frequently used for short-range point-to-point navigation and terminal phase operation in a localized (e.g., battlefield or theatre) area. These are also referenced to earth-fixed, but <u>not</u> generally earth-centered frames, which are in turn tied to specific ground points (e.g., a bombing target or a landing field). Typical of such coordinate systems are UTM coordinates, and tangent-plane, target-centered coordinates.

TABLE I. APPLICABLE MISSION SUMMARY

	MISSION TYPE		TYPICAL AIRCRAFT	MISSION HISTORY: EN ROUTE PHASE: PROFILE	COORD.	ACC'Y	OBJECTIVE AREA PHASE: PROFILE	COORD.	ACC'Y
MILITARY: STRATEGIC	WEAPON DELIVERY		B1	HICR, DESC	GEOD.	MOD	LOWD(TF), PPUP, WPNL	GEOD.	H
MILITARY: STRATEGIC	LOGISTICS TRANSPORT		C-5A	HICR, DESC	GEOD.	MOD	LOWA, ADRP OR LNDG	LCL	H TO VH
MILITARY: STRATEGIC	RECONNAISSANCE		SR-71	HICR	GEOD.	MOD	HICR	GEOD.	H TO VH
MILITARY: STRATEGIC	GEN. PURP. PATROL		P-3C	MECR	GEOD.	MOD	DESC, WPNH	GEOD.	MOD
MILITARY: STRATEGIC	ASW		S-3A	MECR	GEOD.	MOD	DESC, ADRP, WPNL	LCL	H TO VH
MILITARY: STRATEGIC	WARNING/CONTROL		E2A	HICR	GEOD.	MOD	HICR	GEOD.	MOD
MILITARY: STRATEGIC	MPNG/SRVY/OCNGR.SUP.		RC-135	HICR	GEOD.	MOD	MECR	GEOD.	H TO VH
MILITARY: STRATEGIC	WEATHER RECON.		C-130	HICR	GEOD.	MOD	HICR	GEOD.	MOD
MILITARY: STRATEGIC	IN-FLIGHT REFUELING		KC-135	HICR	GEOD.	MOD	HICR	GEOD.	MOD
MILITARY: STRATEGIC	SEARCH/RESCUE		HC-130H	MECR, DESC	GEOD.	MOD	LOWA, ADRP	GEOD.	MOD TO H
MILITARY: TACTICAL	WPN DLVY	CLOSE AIR SUP.	A-7	LOCR	LCL	MOD	DESC, WPNH	LCL	H TO VH
MILITARY: TACTICAL	WPN DLVY	STR./INTERD'N	F-111	MECR	LCL	MOD	DESC, WPNH	LCL	MOD TO VH
MILITARY: TACTICAL	EQPT/PERS. DLY/EXT	ASSAULT	CH-3E	LOCR	LCL	MOD	DESC, LNDG	LCL	H
MILITARY: TACTICAL	EQPT/PERS. DLY/EXT	INSERT/EXTR'N	UH-1	LOCR	LCL	MOD	DESC., HOVR	LCL	H TO VH
MILITARY: TACTICAL	EQPT/PERS. DLY/EXT	FERRY	C-7A	MECR	LCL	MOD	DESC, LNDG	LCL	H
MILITARY: TACTICAL	RECON/OBSERV'N		OV-1/10	LOCR	LCL	MOD	LOWA	LCL	MOD TO H
MILITARY: TACTICAL	SEARCH/RESCUE		HU-16A/E	LOCR, DESC	LCL	MOD	LOWA, LNDG	LCL	MOD TO H
NON-MILITARY	CARGO/PASS. TRNSPRT		747	HICR, DESC	GEOD.	MOD	LOWA, LNDG	GEOD.	MOD TO VH
NON-MILITARY	OIL/MINERAL SRVY		DC-7	MECR	GEOD.	MOD	LOCR	GEOD.	MOD TO H
NON-MILITARY	MAPPING/OCNGR SUP.		DC-7	HICR	GEOD.	MOD	MECR	GEOD.	MOD TO VH
NON-MILITARY	WEATHER RECON.		LCKHD CONST'N	HICR	GEOD.	MOD	HICR	GEOD.	MOD
NON-MILITARY	SEARCH/RESCUE		GRUMMAN AMPH.	MECR, DESC.	GEOD.	MOD	LOWA, LNDG	GEOD.	MOD TO H
NON-MILITARY	FOREST MANAGEMENT		DC-3	MECR, DESC	LCL	MOD	LOWA, ADRP	LCL	MOD TO H

MNEMONICS KEY: LOCR, MECR, HICR = LOW, MEDIUM, HIGH ALT. CRUISE; DESC = DESCENT; LOWA, LOWD, LOMA = LOW ALT. APPROACH, DASH, MANEUVERING; WPNL, WPNH = LOW-G, HIGH-G WEAPON DELIVERY RUN AND RELEASE; PPUP = POP UP, ADRP = AIR DROP, LNDG = LANDING, TF = TERRAIN FOLLOWING; GEOD. = GEODETIC-REFERENCED; LCL = LOCALLY-REFERENCED, MOD = MODERATE; H = HIGH; VH = VERY HIGH.

Each of these coordinate systems has its peculiar advantages and disadvantages. For example, the latitude-longitude approach leads to the simplest navigation equations, but is useless near the poles, while the local vertical direction cosine approach is costlier to mechanize, but automatically provides polar navigation capability. Thus, if mission operations were, say, limited to non-polar regions, the former might be selected. However, if the objective area mission involved, say, an air drop with respect to a target-referenced aim point, then a local coordinate system would be more suitable and natural for navigation in this phase of the mission.

Typical solutions to this problem -- the necessity for navigating in one coordinate system enroute, and another in the objective area -- have often been brute-force; i.e., two different sets of navigation equations are mechanized, one for each phase, and <u>both</u> sets of computations are run in parallel during the terminal phase. This approach tends to be costly in terms of airborne computer loading.

These mission characteristics and requirements -- i.e., accuracy, flight profile, and coordinate reference frame -- constitute important constraints on the design of a generalized processor which should if possible accommodate the entire range of these requirements. The ideas underlying the solutions to these problems are discussed in subsection 3, which deals with basic processor structural and operational concepts.

Determination of the initial processor navigation equipment applicability list was governed mainly by the desire for full processor coverage, with due regard to both accuracy and cost requirements, of the range of missions summarized in Table I. The resulting list therefore included:

a. IMU (rotationally isolated or strapdown)

b. AHRU

c. CADS

d. Radio Transceiver(s) (earth mode or LOS, one way or two way)

In addition, although specific processing capability has not to date been included for the following equipment types, careful design provision has nevertheless been made so that their processing can easily and naturally be accommodated by simply adding for each, its appropriate hardware-specific module:

e. Doppler radar

f. Angular tracking equipment (stellar, optical, etc.)

g. Terrain matching equipment.

3. BASIC STRUCTURAL AND OPERATIONAL CONCEPTS

The attainment of a compact, efficient, and flexible processor design, in the face of the desired broad range of mission and hardware applicability outlined above, required early Phase I consideration of a variety of basic design factors which are discussed in paragraphs a. through e. below. Against this background, paragraph f. then presents a brief summary of the modular organization actually adopted for the processor. Finally, paragraphs g. and h. deal respectively with certain capabilities of special interest which were developed and embedded in the processor design during the program, and a brief discussion of the growth potential also embedded in the processor structure, with regard to expanding its avionics hardware applicability.

a. Basic Navigation Equations

As discussed above, four basic navigation entities -- vehicle 3-d position, vehicle 3-d velocity, vehicle angular orientation, and vehicle angular rate -- were identified early in the program as comprising the set of fundamental outputs which should be generated in the basic navigation implemented by the processor.

Two fundamental and closely related questions immediately arise in this connection. First, what coordinates should be selected to represent each of these entities, and second, to what reference frames should these coordinates be referenced?

Addressing the latter question first, there are in all seven widely used and convenient reference frames which are pertinent to the discussion here. These are the inertial (I), earth fixed (E or EF), locally level (L), air (A), platform (P), and computational (C) frames. These frames are defined in detail in the List of Symbols and Abbreviations. The first five of these frames (I, E, EF, L, and A) comprise the set of candidate frames from which the mechanized C and P frame orientations were selected.

The fact that position and velocity with respect to the earth are two of the fundamental output requirements for the basic navigation processor allowed immediate elimination of the A and I frames in favor of the closely earth-related E, EF, and L frames. Immediate elimination of the A frame (for rotationally isolated platforms) as a P frame candidate was also natural, since this would be tantamount to unnecessarily mechanizing a strapdown platform.

Table II was constructed to aid in completing the C and P frame selections.* The table summarizes the broad computational requirements imposed by any combination of C and P frame selections from among the remaining candidate frames. Overall computational requirements are broken down into 13 different

*Table II is a comparison based on inertial navigation only (see Appendix III). However, since this mode imposes a greater computational load than the others (e.g., ADR) required of the processor, it was felt this was appropriate for a limited tradeoff investigation.

TABLE II. BASIC INERTIAL NAVIGATION COMPUTATIONAL REQUIREMENTS BY P AND C FRAME TYPE

COMPUTAT'L OPERATION	GENERAL MATHEMATICAL DESCRIPTION	EQUATION SIMPLIFICATIONS FOR:							
		EARTH-FIXED C FRAME				LOCAL LEVEL C FRAME			
		GMBLD (OR FLTD) PLATFORM				GMBLD (OR FLTD) PLATFORM			
		LOCAL LEVEL	EARTH-STAB'ZED	INERT. STABZD	STRAPDOWN PLATFORM	LOCAL LEVEL	EARTH-STAB'ZED	INERTIALLY STAB'ZED	STRAPDOWN PLATFORM
ACCELEROMETER CALIBRATION COMPUTATIONS	$f_p = f_{acc} + \Delta f\left(f_p, (\omega_{p/I})_p, \text{CALIBR'N CONST'TS}\right)$	$(\omega_{p/I})_p$ EFFECTS NEGLIGIBLE			$(\omega_{p/I})_p$ EFFC. SIGNIF	$(\omega_{p/I})_p$ EFFECTS NEGLIGIBLE			$(\omega_{p/I})_p$ EFFECTS SIGNIF'T
GRAVITY COMPUTATIONS	$g_c = g_c\,(P_c)$	3 GRAVITY COMPONENTS REQUIRED				ONLY [illegible] GRAVITY COM[illegible] REQ'D			
VELOCITY UPDATE	$\Delta_c V_c = \int_{\Delta t} \left[T_{c/p} f_p + g_c - \left[2T_{c/E}(\omega_{E/I})_E + (\omega_{c/E})_c\right] X V_c\right] dt$	$\omega_{c/E} = 0$, $2(\omega_{E/I})_c$ = CONSTANT VECTOR				REQ'D AS SHOWN			
CARTESIAN POSITION UPDATE	$\Delta_c r_c = \int_{\Delta t} V_c\, dt$	REQ'D AS SHOWN				NOT REQ'D			
ALTITUDE UPDATE	$\Delta_c h_c = \int_{\Delta t} V_{cI}\, dt$	NOT REQUIRED				REQ'D AS SHOWN			
C/E FRAME ANGULAR RATE COMPUTATIONS	$(\omega_{c/E})_c = K_L V_c$	NOT REQ'D: $(\omega_{c/E})_c = 0$				REQ'D AS SHOWN			
C/E FRAME TRANSFORMATION UPDATE	$\Delta_c T_{c/E} = -\int_{\Delta t} \left[(\omega_{c/E})_c X\right] T_{c/E}\, dt$	NOT REQ'D: $T_{c/E}$ = CONSTANT MATRIX				REQ'D AS SHOWN			
C/P FRAME ANGULAR RATE COMPUTATIONS	$(\omega_{p/c})_c = T_{c/p}(\omega_{p/I})_p - (\omega_{c/E})_c - T_{c/E}(\omega_{E/I})_E$	$\omega_{c/E} = 0$; $(\omega_{E/I})_c$ = CONST. VECT. ϕ	NOT REQ'D; $\omega_{p/c} = 0$	$\omega_{c/E} = 0$; $(\omega_{E/I})_c$ = CONST VECTOR		NOT REQD $\omega_{p/c} = 0$*	REQ'D AS SHOWN		
C/P FRAME TRANSFORMATION UPDATE	$\Delta_c T_{c/p} = \int_{\Delta t} \left[(\omega_{p/c})_c X\right] T_{c/p}\, dt$	REQ'D AS SHOWN	NOT REQ'D; $T_{c/p}$ = CONST. MTRX	REQ'D AS SHOWN		NOT REQD $T_{c/p}$ = CON MTRX*	REQ'D AS SHOWN		
BASIC PLATFORM CONTROL RATE COMPUTATIONS	$(\omega_{p/I})_p = T_{p/c}\left[(\omega_{p/c})_c + (\omega_{c/E})_c + T_{c/E}(\omega_{E/I})_E\right]$	$\omega_{c/E} = 0$	$(\omega_{p/I})_p$ = CONST. VCTR	NOT REQD $\omega_{p/I} = 0$	NOT REQD AVAIL FM PLATF. GYROS	$\omega_{p/c} = 0$	REQ'D AS SHOWN	NOT REQ'D $\omega_{p/I} = 0$	NOT REQ'D AVAILABLE FROM PLAT. GYROS
GYRO CALIBRATION COMPUTATIONS	$\omega_{GYR} + (\omega_{p/I})_p + \Delta\omega\left(f_p, (\omega_{p/I})_p, \text{CAL'N CONST}\right)$	f_p AND $(\omega_{p/I})_p$ EFFECTS MODERATE			EFFECTS SIGNIF'T	f_p AND $(\omega_{p/I})_p$ EFFECTS MODERATE			EFFECTS SIGNI'T
PLATFORM ATTITUDE READOUT CAL'N COMPUTATIONS	a	REQ'D			NOT REQ'D	REQ'D			NOT REQ'D
A/C ATTITUDE/ ATTITUDE RATE OUTPUT GENER'N	a	REQ'D			NOT REQ'D; AVAILABLE AS $T_{c/p}$	REQ'D			NOT REQ'D; AVAILABLE AS $T_{c/p}$

* ASSUMES NO RELATIVE TORQUING OF P W/R TO C FRAME

ϕ $(\omega_{p/c})_c = T_{c/p} K_L T_{p/c} V_c$

a. THESE COMPUTATIONS DEPEND ON PLATFORM AND PLATFORM ANGLE AND ANGLE RATE MEASUREMENT EQUIPMENT TYPES.

11123

basic computational operations. Each of these is expressed in terms of a generalized, all-frame, vector/matrix equation (see Appendices II and III), and the simplifications -- if any -- of this operation which result from any particular C/P selection are indicated.

Using the table, it is evident that the choice C = L would be inadvisable relative to the selected C = E/EF, since (a) C = L does not provide E/EF frame-referenced position and velocity as natural outputs, while C = E/EF does; (b) the L frame does rotate (as the vehicle moves over the surface of the earth) with respect to the earth, while the E/EF frame does not; this requires the dynamic updating of the C versus E frame transformation in the former case, but obviates it in the latter; and, (c) most tactical, target-referenced frames are basically E/EF, rather than L frames. The choice C = E/EF has therefore been made for these reasons.

On the other hand, the choice P = L, rather than P = E/EF, seems a natural one for (rotationally isolated) platform attitude stabilization. This is so because (a) it eliminates tumbling of the gravity vector with respect to the inertial instruments; such tumbling excites g-sensitive inertial instrument errors, which can lead to the need for costly modeling of such errors in the Kalman filter; and (b) it provides aircraft roll, pitch and yaw as natural gimbal angle outputs, without the need for further coordinate conversion. However, on closer scrutiny, there are many situations (e.g., during alignment) in which the platform, even though nominally locally level stabilized, will in fact be significantly misaligned with respect to the true L frame. Recognition of this fact -- i.e., by allowing for separate P and L frames in the processor mechanization -- therefore provides a desirable flexibility (and performance improvement) of the processor with regard to the processing of platform instrument inputs and control outputs. In addition, the facts that (a) the L frame is a natural one in which to represent wind estimates, and (b) P and L frame separation is mandatory in the case of a strapdown platform anyway, add force to the argument in favor of such a separation.*

Choosing C and P to be noncoincident does, of course, lead to the computational need for maintaining a dynamic transformation between these two frames. Although this would be unnecessary if the two frames had been chosen to coincide (i.e., by choosing both C and P as L frames, or both as E/EF frames), it is nevertheless felt that the advantages of frame separation, as discussed above, outweigh this and other disadvantages.

The selection C = E/EF also benefits from the fact that the E and EF frames are closely related; i.e., both are fixed to the earth and therefore are not rotating with respect to one another. This has been discovered to result in a very high degree of similarity between the basic inertial navigation and navigation error equations formulated for each of the two frames.

---

*When such a separation is mechanized, an additional interframe transformation to those shown on Table II is of course required. However, since this additive requirement is common to all P frame choices shown, it does not alter the tradeoff comparisons.

As a matter of fact, it has allowed the formulation of essentially a single set of such equations which is equally applicable to E or EF frame-referenced computations, if augmented by some simple, interframe switching computations. This is felt to be a significant advantage with regard to simplification of the overall processor design in multiphase missions.

Turning to the remaining question of coordinate type (e.g., polar, spherical, Cartesian, etc.) the choice of Cartesian coordinates is a natural one, since (a) such coordinates inherently lend themselves to Cartesian, orthogonal vector representation, and are thus compatible with the extensive and advantageous employment of vector-matrix algorithm formulations throughout the processor (see paragraph b. below), and (b) use of these coordinates leads naturally to the selection of corresponding Cartesian coordinates as error state variables in the processor Kalman filter (this produces truly linear measurement-state relationships, which uniquely enable the accurate generation of large Kalman filter error estimates and resultant processor variable corrections -- which would be impossible if the often-used quasilinear angular error variables were employed instead).

b. Vector/Matrix Algorithms

It is evident from Table II that the use of vector/matrix notation greatly simplifies both the overall representation of the equations shown and the identification of common subroutine candidates (e.g., 3x3 matrix multiplication) among those equations as well. Both of these characteristics are patently important advantages in the organization of an efficient, compact, and flexible computer program.

Although Table II is limited only to inertial navigation, almost all other processor functional areas can be similarly formulated. For example, in the area of radio navigation data processing, the fundamental, geometric pseudoranging process [which comprises the whole range of phase/phase difference techniques including Loran, Omega, NAVSAT, TACAN (DME), etc.,] intrinsically always involves -- and therefore can be formulated in terms of -- the vector dot product, cross product and absolute value operations. In the area of airspeed dead reckoning (ADR), wind and airspeed, and their subsequent processing into position and velocity vectors, can also naturally be expressed as vectors and vector-matrix operations, respectively. Most other, non-Kalman operations can also be formulated equally easily and completely in vector/matrix terms.

Finally, in the area of Kalman filtering operations, which can comprise a very large portion of the overall processor computational load in any specific application, use of vector/matrix operations is perhaps the most natural and advantageous of all. This is partly because the equations were originally naturally formulated in vector-matrix terms by Kalman himself, and partly because further, this formulation leads naturally to a highly useful decomposition of the overall filter equations into a set of mainly equipment-dedicated equations, based on the use of standard matrix-partitioning techniques on the unpartitioned equations.

c. Functional Modularity/Commonality

Simple logic dictates that if processor design is to be kept compact and easily manageable, and at the same time capable of accommodating a wide variety of mission and navigation hardware requirements, the ancient but powerful concepts of functional modularity and functional commonality must be exploited. To this end, a careful consideration of how best to decompose the overall basic navigation computations into a set of functionally distinct modules was undertaken early in the Phase I effort.

Two basic ground rules were quickly identified. First, the primary modular division ought to be largely navigation hardware oriented; e.g., all computations associated with IMU data use and control should be grouped into a single, functional, IMU module (or submodule). This would (a) simplify the complete bypassing of such computations as a group in cases of IMU hardware malfunction in an overall hardware configuration containing an IMU, and (b) allow for the easy omission of an IMU-related computation in configuring software for systems not involving an IMU, or for later augmenting the software if an IMU were added.

Second, it was also apparent that within each module of such an overall modular structure, two classes of functional operations could be usefully distinguished: those which were independent of differences in, and therefore common to, the various types of hardware associated with that module, and those which depended on the peculiarities of each such hardware type. For example, assuming a single computational reference frame, there are several well-known mathematical techniques for updating the required IMU platform/computational axes transformation, each one of which could be used for either the strapdown or the rotationally free types of IMU. On the other hand, the algorithms for airborne IMU error compensation for ground-calibrated IMU error sources are highly dependent in form and extent on the nature and arrangement of the IMU gyros and accelerometers.

Use of these two principal ground rules led to the overall modular processor framework described in general by Figure 3, and discussed in paragraph f.

In the search for algorithm commonality, it quickly became evident that the use of vector-matrix algorithm formulations was the key. When expressed this way -- instead of in terms of the customary scalar equations -- superficially disparate types of inertial navigation schemes (e.g., north slaved, free azimuth, strapdown, unipole, etc.) look in large part suspiciously similar. This is no accident, but is due to the simple fact that each of these is essentially a mechanization of the same physical problem; namely (using inertial navigation as an example), the continuous determination of position and velocity by means of a device (the IMU) which measures acceleration (except for gravity), and attitude rate. In addition, Kalman filtering equations (see paragraph d.) are by far most easily and naturally expressed in vector-matrix form. Also, the extensive, well-known partitioning techniques

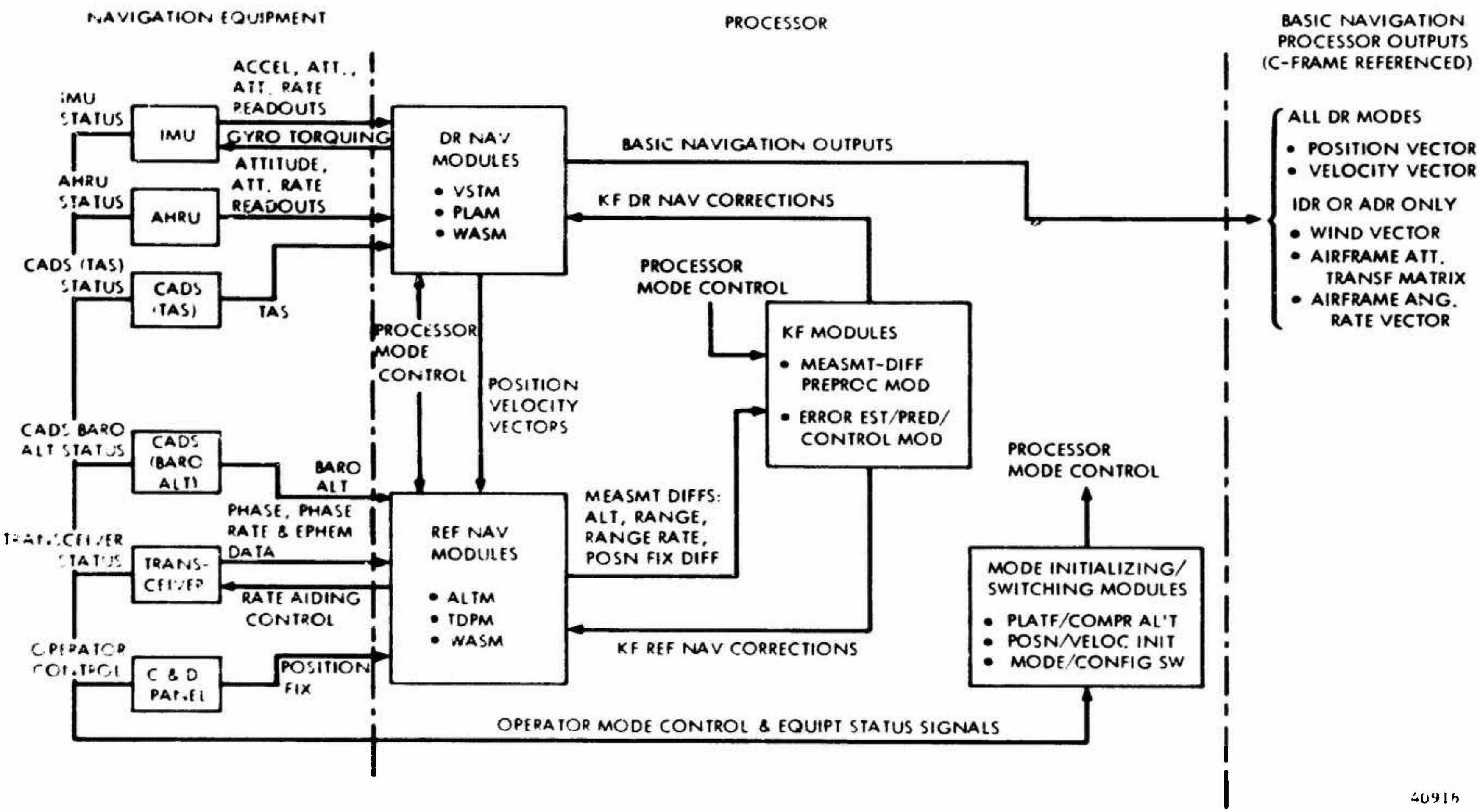


Figure 3. Navigation Processor/Navigation Equipment Overall Organization and Interfaces

associated with vector-matrix formulations lend themselves naturally to the functional modularizing (functional partitioning) process outlined above.

Finally, it is important to emphasize here that the overall functional modularity/commonality structure adopted for the processor under consideration here is principally a means to achieve one of its central and most important characteristics: generality. While the flexibility inherent in this structure guarantees that it will provide easy adaptability to the mission requirements and hardware associated with any single system, and minimize redundant software development for the class of systems to which it applies, it cannot result in software quite as efficient for any single system as that developed specifically for that system. Averaged over many systems, however, generalized software of this type is unquestionably far more cost-effective than the much larger body consisting of the sum of all the specialized software separately developed for each such system.

d. <u>Kalman Filtering</u>

Use of a Kalman filter as an integral and central processor feature was decided upon because: (a) its natural formulation is a linear, vector-matrix one, so that it lends itself to modularization and algorithm commonality with the rest of the processor computations; (b) in the area of pseudoranging in particular, it inherently incorporates the geometric, coordinate-conversion function; (c) it offers the greatest possible theoretical accuracy potential in use of available navigation data, and (d) it provides a natural basis for statistically optimal, emitter data reasonableness and selection algorithms.

Taking these in turn, one of the central conceptual and computational entities in the structure of any Kalman filter is the state estimate vector, or more exactly in the case of vehicle-borne navigation systems, the error state estimate vector. This is simply the set of estimates of the set of scalars which are used to model the errors in the overall navigation system information flow. The entire vector-matrix structure of overall filter computations is based in turn on the order and meaning of these error variables. Once the concept of hardware-oriented modularization of the non-Kalman navigation computations described above was formulated, it became obvious that this concept could be naturally and easily extended to correspondingly modularize the Kalman filtering computations as well. This consisted simply in ordering and grouping the error variables in the overall state vector in such a way as to form a set of separate blocks of variables, each block consisting solely of the set of modeled errors associated with the computations and hardware of just one of the non-Kalman functional modules. This idea was carefully developed, and has been incorporated into the processor to allow for the easy, hardware-dedicated structuring of the Kalman as well as the non-Kalman portions of the overall processor software.

As mentioned above, the Kalman filter basically generates continuous estimates not of the overall navigation variables themselves, but only of the errors in these variables. These error estimates are then used to periodically correct the values of the overall navigation variables. Correspondingly, these error estimates are generated by use of (noisy) measurements of the differences between two synchronous estimates of some geometric quantity, so that, since the true value of this quantity cancels out in the differencing, the measurement really consists of the difference of the errors in the two estimates of the quantity. The measurement difference is therefore directly related to certain of the primary navigation error variables (i.e., state variables). If the errors are small, then this in general nonlinear relation* can be approximated by a linear one (the so-called Kalman measurement or observation matrix), which embeds the coordinate conversion function. For example, if the geometric quantity being estimated is range to a simple emitter, the (say) two-way signal phase shift provides one range estimate, while use of Pythagoras' theorem on user and emitter position coordinates provides another. In differencing, the true range cancels out, so that the measurement difference really involves only the difference of the user and emitter location errors along the line of sight, plus of course the inevitable ranging signal phase, propagation and receiver noise errors. Assuming that the emitter location error is negligible, the filter will in fact estimate the component of user position error along the user-emitter line of sight, using the user-emitter geometric information embedded in the measurement matrix, and will then correct user position accordingly. Thus not only does the filter embed the necessary geometric coordinate-conversion, but it can make maximum use of radio navigation information data (e.g., range) from whatever number of emitters whose signals are currently available, even if this is less than (or more than) the number required to determine, say, 3-d user position.

In addition to these coordinate conversion capabilities, the Kalman filter is also a minimum-variance, statistical filter. That is, given a measurement difference, its consequent estimates of error state variables are based on (in addition to the linear geometric relations discussed above), the relative uncertainty it associates with the variables concerned. Specifically, these uncertainties are carried as numerical variances, one associated with each state variable. For example, if in the ranging example above, the variance associated with the user position error was much larger than that associated with the emitter, then the filter would assign the entire measurement difference to user position error, and reset it alone. On the other hand, if the converse were true, only the emitter position error would be estimated and only the emitter position error reset accordingly. For intermediate cases, both would be partially reset. This kind of filter is therefore optimum in the sense that, given a large number of noisy measurements,

*If the relationship is actually linear, then neither the errors nor the measurement-differences need be small.

and accurate error variable statistics, it will theoretically reset processor navigation variables with less residual error than any other type of filter.

Finally, the error estimate variances carried by the filter to accomplish the minimum-variance weighting described above, also provided a natural and convenient basis for the construction of statistical, emitter data reasonableness and selection algorithms, which are embedded in the current processor definition. These are discussed in paragraph f.

e. Overall Navigation Information Processing Organization

Once a set of navigation system sensor hardware has been decided upon, a range of options remains as to the manner in which the software shall process (i.e., conduct navigation using) the data available from these sensors. At first sight, all these options would seem to fall into one of two familiar classes; i.e., the use of the data from any one given sensor either (a) to directly drive DR (dead reckoning) navigation in the intervals between successive availability of reference measurement data, or (b) to provide such periodic, reference measurement data itself.

For example, assuming an equipment set consisting of an IMU and a NAVSAT receiver, a natural data processing philosophy would be to use the essentially continuous IMU accelerometer data to continuously (i.e., at a very high data rate) and directly drive the velocity and position updates, and to use the (lower data rate) NAVSAT pseudoranging data as reference measurements to periodically correct---via an appropriate navigation filter ---DR position and velocity, as well as (perhaps) the errors in IMU measurements of acceleration, and the errors in NAVSAT pseudoranging measurements due to NAVSAT/receiver clock phase difference and signal propagation delay errors.

On the other hand, assuming the availability of an adequately high receiver pseudoranging data rate, then position (and perhaps velocity as well if doppler shift data is mechanized) might be directly tracked by use of the NAVSAT receiver outputs*, and the (lower data rate) IMU data treated as navigation reference data for processing through the navigation filter.

This latter approach (NAVSAT dead reckoning) is less desirable than the former (IMU dead reckoning), however, because the former is essentially a differential (predictive), and the latter an integral (historical) process. That is, when the NAVSAT DR approach is used, vehicle acceleration can only be inferred by noisy differentiation of the DR position and/or velocity estimate, while in the IMU DR (IDR) approach, it is measured directly by the IMU accelerometers, and velocity and position are inferred by noise-smoothing integrations.

*Simultaneous three-channel tracking of at least three satellite emitters would of course be necessary to maintain 3-d position and velocity.

When no IMU is available [or no AHRU/TAS backup equipment combination to provide (integral) airspeed dead reckoning (ADR) instead], so that position-level DR via NAVSAT or other pseudoranging-type equipment is necessary, then this technique can be improved upon—i.e., made more integro-historical —by creating an artificial computational model of vehicle dynamics (acceleration, velocity, and position) using whatever is known (either a priori or via other onboard equipment) about the dynamical characteristics of the carrier vehicle, to implement this model. For example, in a coordinated turn, the vehicle acceleration vector is always normal to its velocity vector, and the occurrence of such a turn is signalled by the behavior of the vehicle control surface settings. Such information could be used to condition an appropriate filter (e.g., a Kalman filter) to "expect" (and therefore to better estimate) a large cross-track acceleration when, say, aileron setting is suddenly changed. This general technique—creating a vehicle dynamics model to accomplish continuous dead reckoning navigation in the intervals between successive use of position measurement data (via say pseudoranging data)—has for convenience of reference been termed pseudo dead reckoning (PDR) and has been tentatively adopted and embedded in the overall processor design, for use whenever no source of continuous acceleration-level or velocity-level data for DR navigation is available.

In particular, this mode can be used to advantage in start-up situations, to maintain continuous radio-autonomous navigation while coarse IMU or AHRU (if available) alignment relative to the computational (C) frame is being carried out.

The altitude channel is also given a similar treatment to provide overall processor uniformity of operation in the face of loss of barometric altitude information. As long as such data is available, it is treated as reference measurement data (regardless of whether the processor DR mode is IDR, ADR, or PDR). When such data is lost, however, the last value of altitude is retained and decayed slowly toward nominal vehicle cruise altitude. This pseudo altitude is, however, still treated as a reference altitude measurement by the Kalman filter (but given a growing statistical uncertainty).

To summarize, for the complete navigation equipment complement—IMU, AHRU, CADS, and transceiver(s)—comprising current processor processing capabilities, the basic processor navigation mode rules are simply:

- If IMU is available, then IDR
- If IMU is not available, but AHRU/CADS(TAS) is, then ADR
- If neither IMU nor AHRU/CADS(TAS) is available, but radio pseudoranging equipment is, then PDR
- Radio pseudoranging and barometric/pseudoaltitude data (or visual position fix data) is always used <u>only</u> as reference navigation measurement data through the Kalman filter, and never as a direct DR data source.

- "Available" in the above means:

  - IMU or AHRU: Coarse platform-to-computer alignment is complete and valid navigation data is being generated.
  - CADS(TAS or barometric altitude): Valid navigation data is being generated.
  - Pseudoranging equipment: Signal acquisition is complete and valid navigation data is being generated.

f. Processor Modular Organization

With the preceding discussion as a background, this subsection presents an overall summary of the modular navigation processor, which is presented in MLI specification-level detail in Section III.

The overall processor has been organized into four conceptually distinct modular groups—the dead reckoning navigation (D or DR) modules, the reference navigation measurement (R) modules, the Kalman filter (K or KF) modules, and the processor mode/configuration initialization and switching modules. These four modular groups are discussed in order in the following paragraphs. As a reference in these discussions, Figure 3 summarizes information flow between these modular groups within the processor, and across the processor/navigation sensor interface as well.

(1) DR Nav Modules

This group includes the Vehicle State Module (VSTM), the Platform Module (PLAM), and the Wind/Airspeed Module (WASM). Together, these modules comprise the processing necessary to conduct continuous, basic DR navigation in the IDR, ADR, or PDR modes. In all DR modes, this means the continuous generation of the C frame referenced position and velocity vectors. In the IDR or ADR modes, where a platform is available to measure airframe attitude (and perhaps attitude rate), the orthogonal airframe-to-(earth-fixed) computational frame, and the corresponding C frame-referenced airframe angular rate vector are additionally generated as basic navigation outputs (and a wind vector as well). Table III summarizes the use and function of each D module by DR mode.

From Table III it is evident that a considerable degree of intermodal functional commonality exists. This has been emphasized in the MLI specification for each module by grouping (and logically addressing) the overall functions required of each module into those required in common for all three DR modes, those required in common only for modes in pairs, and those particular to each mode. This grouping, together with the simple DR mode-to-DR navigation hardware sensor correspondence rules (see Section III, paragraph 3.e), has resulted in a very significant simplification of the processor DR mode switching logic, and in a very simple and direct DR nav hardware-to-software modular correspondence. These techniques have also been used in both the R module and K module groups, discussed in paragraphs (2) and (3).

TABLE III. DR NAV MODULES: SUMMARY OF OPERATIONS AND USE BY MODE

<table>
<tr><th>DR Nav Mode / DR Module</th><th>IDR
(With CADS TAS)</th><th>ADR</th><th>PDR</th></tr>
<tr><td>VSTM</td><td>• Processes PLAM P Frame specific force into C Frame velocity and position vectors</td><td>• Processes C Frame WASM airspeed and wind vectors into C Frame velocity and position vectors</td><td>• Generates L Frame pseudoacceleration vector
• Process this into C Frame position and velocity vectors</td></tr>
<tr><td rowspan="2">PLAM</td><td colspan="2">• Generates L/C, P/L, P/C, A/P interframe transformations and angular rate vectors for VSTM, WASM, PLAM, and processor output use</td><td rowspan="2">• Not required</td></tr>
<tr><td>• Error-compensates IMU gyro, accel'r, attitude, attitude rate readouts</td><td>• Error-compensates AHRU, attitude, attitude rate readouts</td></tr>
<tr><td>WASM</td><td>• Error-compensates and resolves airspeed into C Frame and subtracts from VSTM velocity to obtain wind vector</td><td>• Error-compensates and resolves wind and airspeed into C Frame vectors for VSTM ADR operation</td><td>• Not required</td></tr>
</table>

From Table III it is also evident that a high degree of interdependence exists between the separate DR modules. This, as will be seen, is in direct contrast to the R modules, which are essentially mutually independent.

(2) Reference Navigation Measurement Modules

This group includes the reference altitude module (ALTM), the position fix module (POSM), and the transceiver data processing module (TDPM). In general, each of these hardware-specific modules executes two types of operation. First, it accomplishes the processing of input sensor data from its particular sensor or input source (and in the case of the TDPM, provides rate aiding control feedback as well) into an estimate of some fundamental navigation quantity (e.g., barometric altitude, radio pseudorange). Second, it synchronously computes a second estimate of the same navigation quantity based on the basic DR navigation module outputs and forms the synchronous difference. These synchronous differences form the basic input measurements used by the Kalman filter to subsequently estimate and correct D and R module-associated navigation quantities. Since these R modules are mainly independent of one another, they are discussed in turn in the following paragraphs.

(a) Reference Altitude Module (ALTM)

The functions executed by this module depend on whether the vehicle is on the ground or airborne, and whether or not barometric altitude is available, as summarized in Table IV. Note that synchronous differencing of the DR and reference altitude measurements is specified in order to remove vehicle dynamics. This technique is uniformly specified for all R modules.

(b) Position Fix Module (POSM)

This module simply converts panel-inserted visual position fix data from input coordinates (e.g., latitude/longitude/altitude) into an internal C frame-referenced position vector, and synchronously differences this vector with the corresponding DR-generated position vector.

(c) Transceiver Data Processing Modules (TDPMs)

The design technique utilized in establishing the overall TDPM module group was a detailed examination of the functional tasks required for the various emitter types. Three basic emitter/user transceiver configurations were identified by this effort: the Ground, the Airborne, and the Satellite Emitter configurations. A set of overall, all-configuration functional tasks was established and organized into a corresponding set of TDPM modules. Within each module the attendant functional tasks were then further organized in terms of their use by configuration type.

TABLE IV. ALTM: SUMMARY OF OPERATIONS AND USE BY REFERENCE ALTITUDE MODE

CADS Baro Altitude Availability \ Aircraft Status	Air	Ground
Available	• Ref Alt = Error Compensated Baro Alt • Differences** DR/Ref Alt	• Baro Alt = Field Alt* • Ref Alt = Field Alt* • Differences DR/Ref Alt
Not Available	• Ref Alt = Pseudo Alt • Decays Pseudo Alt • Differences DR/Ref Alt	• Ref Alt = Field Alt • Differences DR/Ref Alt

*Panel-inserted field altitude assumed available.
**Synchronous differencing

The modules established by this functional organization were:

- Emitter Word Processing (TEWM)
- Propagation Correction (TPCM)
- Range and Range Rate Generation (TRRM)
- Kalman Measurement Observables (TMOM)
- Antenna Lever Arm Compensation (TALM)
- Kalman Measurement Matrix (TMMM)
- Data Statistics Generation (TDSM)
- Acquisition and Aiding (TAAM)

An important design feature which has been incorporated in the module definition is the combination of both range and range rate calculations within the same modules, since these entail almost identical mathematical operations and from a programming structure standpoint could therefore share the same operational subroutines with only a variation of input and output parameter definition. The design also embeds a single propagation link error state definition which represents the amalgamated effect of numerous algorithmic uncertainties in determining the exact link error. The inclusion of an expanded error state vector of increased dimension was not considered as being warranted due to the basic unobservability of the various link error contributors.

It should be noted that the radio navigation treatment in this document is almost exclusively LOS-oriented, since attention in Phase II was concentrated on this type, rather than on the earth mode type, of radio signal. However, a brief joint Northrop/Magnavox review of the LOS radio navigation module specifications presented in subsection III.3.c of this report revealed that five of those (the TEWM, TALM, TRRM, TMOM, and TMMM modules) would require only minor revision to additionally accommodate earth mode emitter processing, two others (TDSM and TAAM) would need careful review to ascertain how much modification would be necessary, and only one (TPCM) would clearly require major extension.

(3) Kalman Filter Modules

There are eight Kalman Filter (K) modules, falling into four distinct conceptual groups as follows:

- Prediction Modules: Estimate/covariance Matrix Time Update Module (KTUM), and the Time Update Matrix Generation Module (KTMM)

- Measurement Preprocessing Modules: Measurement Matrix Generation Module (KMMM), and the Measurement Reasonableness, Combination, and Optimal Selection Modules (KMRM, KMCM, KMOM)

- Filtering Module: Estimate/Covariance Matrix Filtering Update Module (KFIM)

- Control Module: Estimate/Processor Control Module (KCOM).

The functions of these modules are best understood against the background of the fact that, because of the extent of the computations required to fully execute one cycle of all Kalman operations even when only a small navigation system error model is incorporated in the filter design, the Kalman execution cycle is of necessity considerably longer than either the DR module execution cycles or the even slower R module cycles. Typical values, for example, might be a 0.1-second DR cycle, a 1.0-second R cycle, and a 10-second KF cycle.

In order to accurately track significant system time-variable error dynamics (by computing their effect after the fact) the error estimate produced by the filter is purposely made to lag essentially one KF cycle behind the real-time error behaviour. The control fed back to other processor modules by the filter must therefore be compensated for this lag to avoid the destabilizing effect of delay in the stationary portion of system error dynamics. For these reasons, the various K modules operate asynchronously, some conducting essentially last-cycle-related, and some current-cycle-related, operations.

With this as background, the roles of each of the modules within the above groups during a full-operation Kalman cycle are as follows:

- Prediction Modules:

  KTUM: Time updates KF error estimate and associated covariance matrix across last KF cycle, using time update matrices generated by KTMM in last KF cycle.

  KTMM: Generates current-cycle time update matrices for use by KTUM in next KF cycle.

- Measurement Preprocessing Modules:

  KMMM: Generates current-cycle-measurement/KF cycle endpoint-synchronizing measurement matrices for use by KMRM, KMCM, and KMOM in next KF cycle.

  KMRM: Generates current-cycle statistical reasonableness tests on candidate last-cycle measurements (and their associated matrices).

  KMCM: Combines (linearly or nonlinearly) last-cycle measurements (and their associated matrices) which have passed KMRM reasonableness tests.

  KMOM: Optimally orders current-cycle KMCM output (last-cycle) measurements (and associated matrices) for use by KFIM.

- Filtering Module:

  KFIM: Updates KF error estimate and associated covariance matrix using KMOM outputs (last-cycle measurements and associated matrices). Resulting estimate and covariance matrix relate to start of current KF cycle.

- Control Module:

KCOM: Predicts KFIM estimate to end of current cycle and computes appropriate corrections to DR and R module navigation variables, and to KF estimator. Applies the former to DR and R modules at end of current cycle, and applies estimator correction generated by last-cycle KCOM operation to this-cycle KFIM estimate (start-of-cycle estimate).

Each of the functions executed by each of these modules has been partitioned. That is, instead of algorithms formulated to process the full error state estimate vector (and all its associated vectors and matrices) essentially as a single entity, the algorithms have instead been uniformly formulated to process a set of much smaller function- and hardware module-oriented partitions, which together comprise the equivalent of the full-state algorithms.

There are several levels of this partitioning, as follows. At the broadest level is the partitioning of the overall state x into a DR navigation-related substate, x_D, and a reference navigation-related substate, x_R. In particular, x_D includes all of the elements in the overall mechanized error state model which model errors in the DR navigation module variables, and x_R includes all those elements which model errors in the R module navigation variables.

Each of these principal substates is in turn further partitioned into more specific substates. Specifically, the D substate is broken down into position and velocity error substates which are common to all DR navigation modes (and which model VSTM errors), plus a set of mode-specific substates for each particular DR mode. For example, in IDR, these additional substates would probably include at least a platform-to-computer misalignment substate, and -- depending on the model depth required for the specific processor application -- selected platform drift rate and acceleration measurement error substates (note that these model PLAM errors).

The R substate is correspondingly decomposed into a set of R navigation module-specific substates. For example, the nth emitter net error substate, which models the errors associated with pseudoranging on emitters of a particular net, might range from one including a common user-emitter net clock error substate plus an emitter ephemeris error substate plus separate propagation error substates for each emitter in the net, to no modeling at all -- depending on the model depth required and/or feasible.

Note that, although it is partitioned, the filter is not artificially (and deleteriously) decoupled by simply ignoring the inter-substate covariance matrices, which embed the vital inter-substate correlations that enable cross-correction of errors by measurements which do not directly measure these errors. These matrices are retained and updated so long as both substates concerned are present.

This bi-fold modularity -- via the overall, eight-module Kalman function-specific filter structure, and the uniform, intra-KF-module partitioning into D and R module-specific substate operations -- provides for an extremely flexible, but at the same time highly efficient Kalman filter design. In particular, the problem area of Kalman filter switching -- between DR modes, or of the R configuration (e.g., as emitter nets drop out or are acquired in the course of a mission) -- is dramatically simplified and systematized. Further, the reduction of conceptually difficult Kalman filtering to its essential elements using the above techniques (and at the same time emphasizing its vector-matrix character) will in general result in much more efficient programming, since the programmer can -- after a suitable learning phase -- much more easily visualize filter interrelationships and commonalities.

(4) Initialization/Switching Modules

There are six of these modules, falling into two general classes as follows:

- Initialization Modules: Navigation Start Module (NSTM) and Coarse Align Module (CALM)

- Switching Modules: DR Navigation Switching Module (DSWM), Reference Navigation Measurement Switching Module (RSWM), Kalman Filter Switching Module (DSWM), and C Frame Switching Module (CSWM).

These are discussed in turn in the following paragraphs.

(a) Initialization Modules

These modules are used to establish initial PDR navigation and to initiate preparation for subsequent ADR or IDR operation.

In particular, the NSTM -- which is executed only once, just after the operator navigation start command, and prior to first execution of any other processor module -- simply initializes the C frame to the D frame, generates initial null vector estimates of VSTM position and velocity and of KF position and velocity error substates, and assigns very large initial variances to the latter. Assuming the availability of a panel-entered initial visual fix and/or continuous LOS pseudoranging data, the Kalman filter will subsequently quickly correct position and velocity to navigation-quality accuracies.*

*Vehicle maneuvering is assumed minimal during this period.

The CALM, which is used only when an unaligned platform is available, generates initial values for two of the fundamental interframe matrices -- P/L and L/C -- which are dynamically updated by the PLAM in subsequent ADR or IDR operation. (The third, A/P, is not updated based on prior values, but is directly computed from current dynamic platform attitude readouts.) In addition, the CALM accomplishes any required coarse erection and leveling of a rotationally isolated IMU (this is of course not possible for a strap-down IMU).*

When CALM operation -- which runs in parallel with, and is aided by, reference measurement-augmented PDR -- is complete, the processor is ready for sequencing into ADR or IDR, depending on which platform is available. When both are available, ADR can be initiated and used for DR navigation (using the prealigned AHRU) while IMU alignment is under way.

(b) Switching Modules

These modules are used to switch between DR navigation modes and between reference navigation measurement configurations, depending on navigation hardware availability. In particular the DSWM, RSWM, and KSWM respectively accomplish the inter-DR-mode switching of the DR modules, the inter-R-configuration switching of the R modules, and the corresponding, intra-KF-module D and R substate partitions switching. The CSWM, on the other hand, switches all affected D and R module variables from one C frame to another, when required and specified by the operator. Kalman module variable C frame switching is embedded as a KSWM submodule. The switching modules are discussed in turn in the following paragraphs.

The DSWM accomplishes two main functions. First, it selects the appropriate DR mode based on current DR navigation equipment availability. Second, it appropriately initiates CALM operation if required for aggradation to ADR or IDR, initiates any new-mode-only DR module variables, and sets up intra-DR-module execution of those DR module subsets appropriate to the new mode.

The RSWM also accomplishes two main, corresponding but simpler functions, since the R modules, unlike the D modules, are essentially mutually independent. First, it selects which R modules are to be executed, and in what mode (e.g. reference altitude modes) based on current R navigation equipment availability. Second, it appropriately initiates any new-signal acquisition and tracking if required in the case of radio pseudoranging equipment, initiates any variables required by start or restart of any module, and sets up intra-module execution of the selected R modules in the appropriate modes.

*Only passive processor interfacing with the AHRU is assumed in the current processor definition. The AHRU is assumed to be independently aligned prior to its use by the processor, which at no time generates AHRU torquing rate controls, but only passively uses its attitude inputs in conjunction with TAS to conduct ADR navigation.

The KSWM carries out KF module switching which is compatible and synchronous with both DR navigation mode and R configuration switching. Since such switches can occur at a faster rate in certain situations than the KF cycling rate, the KSWM sets up temporary, mode-change-type-dependent bypassing of certain KF modules conducting lower priority functions, in favor of modules conducting higher priority functions, to conserve time with minimal impact on accuracy. For example, if the switch involves a DR mode change only, then all KF measurement use and control operation modules are temporarily bypassed, to enable the prediction modules (KTMM, KTUM) to maintain error estimate and covariance matrix continuity and currency despite the time-consuming DR error model switching operations which the mode switch requires. On the other hand, if the switch involves only an R configuration change, then only the KCOM needs to be bypassed (to avoid time-consuming adjustment of its prediction operation, since the changes to the measurement use modules are trivial and require little time).

Because of the D and R module-oriented Kalman filter state partitioning, maintaining continuity and currency of the error estimate and covariance matrix at a mode or configuration switch reduces to implementation of the following simple rules:

- Discontinue updating of, and discard all preswitch-only substate estimate vectors and covariance matrix partitions.

- Continue updating (using new-mode updating relationships) of all pre- and postswitch-common, substate estimate vector and covariance matrix partitions.

- Initialize and initiate new-mode updating of all postswitch-only, substate estimate vectors and covariance matrix partitions.

Finally, the CSWM, in conjunction with the Kalman filter C frame switching submodule of the KSWM, switches all DR, R, and KF module variables which are C frame referenced, to their new values with respect to the new C frame. In order to simplify overall processor switching requirements, these variable transformations are delayed until the end of the KF cycle in which the operator has initiated the C frame change, and are then executed in a single pass through the required computations.

g. Special Processor Concepts and Capabilities

This subsection deals with a variety of special concepts and capabilities associated with the processor, which, because they are of unusual interest and value, need emphasis. There are three main areas treated here: (1) switchable uniframe navigation, (2) Kalman filter partitioning, and (3) Kalman filter measurement preprocessing techniques. These are discussed in turn in the following paragraphs.

(1) Switchable Uniframe Navigation

The generalized mission requirements constraints on processor design outlined in subsection 2 above included in particular the need to navigate in any of a variety of either geodetically or locally referenced coordinates on missions of different types, or even in different phases of the same mission.

The approach adopted to simplify and unify processor characteristics in this regard consisted in the selection of a single type of earth-fixed, Cartesian coordinate frame in which the basic navigation computations could be executed, independent of mission type. On missions, or during mission phases, where global navigation is required, this frame is earth fixed and earth centered, with its orthogonal axes aligned along the principal axes of the earth (i.e., polar and equatorial axes); on the other hand, on missions, or during mission phases, where localized navigation is needed, although still earth-fixed, the frame is now centered at some convenient local point (e.g., the touchdown point in ILS), and the axes are aligned in some locally, operationally convenient way (e.g., along, across, and vertical to the runway). A significant property of these frames, since they are nonrotating with respect to one another (both are fixed in the earth), is that the basic navigation and navigation error behavior are almost exactly, functionally the same for both frames. Thus essentially the same navigation and Kalman filter update equations can be used by the processor for global or tactical missions, or, with appropriate (and simple) navigation variable switching at transition, for missions involving both global and tactical phases.

The processor therefore essentially carries out continuous navigation in and with respect to a single type of coordinate system throughout any mission, using essentially a single set of overall navigation computations. User position initialization, emitter position, and other local reference point position and velocity data must of course be transformed from the coordinates in which it is available into this central computational frame, so that all relative navigation and guidance can be carried out uniformly within this frame. The generalized vector-matrix equations which underlie this capability are derived in Appendices II, III, IV, and V.

(2) Kalman Filter Partitioning

The breakdown of the overall state vector into a set of module-related, partitioned substates has been discussed earlier. This outlines the basic concepts of, and advantages provided by the partitioning.

The fundamental unpartitioned (full state) Kalman filtering operations are (a) the time update or prediction operation, (b) the measurement update or filtering operation. Both of these are essentially updating operations on the state vector (x) and its associated covariance matrix (P).

The mathematical, full-state formulations* are:

- Time Update

$$\dot{\phi} = A\phi \qquad (\phi_o = I) \tag{1}$$

$$x = \phi x \tag{2}$$

$$P = \phi P \phi^T + R \tag{3}$$

- Measurement Update

$$b = PM^T/Q \quad (Q = MPM^T + C)^{-1} \tag{4}$$

$$x = x + b\,(Y - Mx) \tag{5}$$

$$P = P - Qbb^T \tag{6}$$

where A is the overall error state differential equation coefficients matrix, ϕ is the error state transition matrix, R is the error state noise matrix, Y is the measurement, M is the measurement matrix, C is the measurement noise, and b is the gain vector.

The partitioning of the full state x is defined by:

$$x = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

where each x_s (s = 1,2...,n) is a module-oriented, partitioned substate of x. The corresponding partitions of A, ϕ, P, R, b, and M (Y and C are assumed scalars) are all double-indexed to indicate the substates they relate; e.g., $A_{ss'}$ is the set of error state differential coefficients which dynamically couple the error substate s' into the error substate s.

*These have been purposely simplified, and simplifying assumptions have also been made about some of the retained auxiliary matrices (i.e., Y and C are scalars), to maintain clarity of ideas.

By the laws of matrix algebra, equations (1) through (6) above can be partitioned into:

$$\dot{\phi}_s = \sum_{i=1}^{n} A_{si}\ \phi_{is} \qquad (\phi_{sso} = I,\ \phi_{ss'o} = 0) \tag{7}$$

$$x_s = \sum_{i=1}^{n} \phi_{si}\ x_{is} \tag{8}$$

$$P_{ss'} = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\phi_{si}\ P_{ij}\ \phi^{T}_{js'}\right) + R_{ss'} \quad \left(P_{s's} = P_{ss'}{}^{T}\right) \tag{9}$$

$$b_s = \sum_{i=1}^{n} P_{si}\ M_i{}^{T}/Q \quad \left(Q = \sum_{i=1}^{n} \sum_{j=1}^{n} M_i\ P_{ij}\ M_j{}^{T} + C\right) \tag{10}$$

$$x_s = x_s + b_s \left(Y - \sum_{i=1}^{n} M_i\ x_i\right) \tag{11}$$

$$P_{ss'} = P_{ss'} - Qb_s\ b_{s'}{}^{T} \qquad \left(P_{s's} = P_{ss'}{}^{T}\right) \tag{12}$$

where s,s', i, and j all range from 1 to n.

The required summations in these equations can either be simplified or in many cases even entirely eliminated by appropriately ordering and organizing the selected substates. For example, if A is organizable into a pseudodiagonal form (this is in fact the case with the combined R substates) then (7) through (9), for the R substates, reduce to:

$$\dot{\phi}_s = A_s\ \phi_s \quad (\phi_{so} = I) \tag{13}$$

$$x_s = \phi_s\ x_s \tag{14}$$

$$P_{ss'} = \phi_s\ P_{ss'}\ \phi_{s'}^{T} \tag{15}$$

For another example, if A is organizable into a pseudo upper-diagonal form (this is in fact the case with the combined D substates in any mode) then, assuming say, that A and ϕ have the partitioned structures:

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ 0 & A_{22} & A_{23} \\ 0 & 0 & A_{33} \end{bmatrix} \qquad \phi * = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ 0 & \phi_{22} & \phi_{23} \\ 0 & 0 & \phi_{33} \end{bmatrix}$$

*It can be straightforwardly demonstrated, using the well-known properties of the transition matrix [all of which are based on equation (1)], that if A has the structure shown, then ϕ must have the structure shown.

then (7) through (9) (for the D substates) would reduce to:

$$\dot{\phi}_{ss'} = \sum_{i=s}^{s'} A_{si}\ \phi_{is'} \quad (s \leq s',\ \phi_{sso} = I_o,\ \phi_{ss'o} = 0) \tag{16}$$

$$x_s = \sum_{i=s}^{n} \phi_{si}\ x_i \tag{17}$$

$$P_{ss'} = \sum_{i=s}^{s'} \sum_{j=1}^{s'} \left(\phi_{si}\ P_{ij}\ \phi^T_{js'}\right) + R_{ss'} \quad \left(s \leq s',\ P_{s's} = P^T_{ss'}\right) \tag{18}$$

In (16), (17), and (18), full advantage has been taken of the structures of A and ϕ, to edit out almost all null ϕ and A submatrices.

In particular, equations (14), (15), (17), and (18) constitute a very attractive basis for the design of the KTUM module. The programmer, who is always primarily concerned with the critical program execution time-versus-storage requirements balance tradeoffs, has at his disposal a set of compact, completely indexed (and indexed in a D and R module-oriented way), already fast (because nearly all null submatrices have already been edited out) equations, in which is embedded a small set of easily identifiable, candidate common vector-matrix subroutines (summation and multiplication). He can optionally -- depending on his requirements and the computer involved -- increase speed by uniformly straightlining, instead of indexing, all indicated operations and subroutines, or at the other extreme he can decrease speed (but reduce storage requirements) by fully indexing all operations. Intermediate decisions result in intermediate speed versus storage consequences.

A similar programming flexibility with regard to the filtering module (KFIM) design is inherent in equations (10) through (12), where in particular an added flexibility in programming is obtained by noting that the indicated summations need be carried out only for the substate indices which have non-null relationships with the measurement Y; i.e., with the non-null submatrices of M. For example, if the measurement Y is a simple two-way range measurement taken synchronously with the KF cycle endpoint, then only the position substate partition of M (which is simply the unit LOS vector transposed) is non-null (assuming that propagation errors are not modeled), and therefore only the index corresponding to the position error substate need be considered in the summations shown in equations (10) and (11).

Correspondingly, equations (13) and (16) furnish an equally attractive starting point for KTMM module algorithm design. Note that, because of the structure of A, the on-(pseudo) diagonal, DR transition submatrix differential equations generated by equation (16) are of the same simple, autonomous form as those of equation (13). This means that <u>independent</u> solutions can be generated for all of the (pseudo) diagonal transition submatrices. Further, careful examination of equation (16) leads to the conclusion that the off-diagonal, substate-coupling transition submatrices can also be separately (but not independently) generated, using the on-diagonal submatrix solutions as forcing functions.

All the non-null partitions of the overall transition matrix can therefore be generated separately, some completely autonomously, others forced by the autonomous solutions. These partitions can then of course serve as direct and natural inputs to KTUM time update operations based on equations (14), (15), (17), and (18). Further, the fact that they can be generated separately allows for the very attractive possibility that each can be updated at a rate consistent with its own dynamics (i.e., its own A matrix). That is, those transition submatrices whose dynamics are essentially constant over one or more KF cycles need be computed only that often, while those whose dynamics are rapidly changing can be updated as fast as necessary within each KF cycle.*

(3) KF Measurement Preprocessing

This paragraph discusses the wide variety of KF measurement preprocessing techniques which have been considered in this development to date. Some of these were and are in common use, while others, to the author's knowledge, are original with Northrop during this development. In any event, all are important design tools, not only for preprocessing the range of reference navigation measurement types which have been considered to date for processor use (i.e., radio pseudorange, altitude, and visual position fix data), and which have largely motivated the development of these techniques, but for other reference measurement sensor types as well (e.g., doppler radar, stellar and landmark trackers, etc.).

There are in all five broad areas considered in the following paragraphs: (a) raw measurement**/KF estimate synchronization (KMMM); (b) time smoothing of the measurements of a single type over all or part of the Kalman cycle (also KMMM); (c) combination of measurements of different types (both linear and nonlinear combination) (KMCM); (d) statistically optimal, ordered selection of measurements of different types (KMOM), and (e) reasonableness testing (KMRM). In general, it is emphasized that whether none, some, or all of these techniques at present should be employed in a given application, and in what order, seems to depend to a large extent on what the mission and hardware background to the particular application is. This whole important area needs further attention to fit the processing tools defined here to the mission and system hardware requirements in as generally applicable a manner as possible.

---

*This attractive approach can in fact be implemented as described if KF measurements are only taken synchronously with the KF cycle endpoints. If measurements interior to the KF cycle are used, however, difficulties arise because partial-cycle transition submatrices may be additionally required to synchronize such measurements with the KF cycle endpoints. In such cases, frequency of generation of even the slow-dynamics dependent transition submatrices (but only those needed for synchronization) may be governed by the measurement data rate. This important area needs further attention.

**I.e., the single sample, mutually synchronous D/R reference measurement/DR differences.

(a) KF Measurement/Estimate Synchronization*(KMEM)

Figure 4 depicts the timing of KF measurements relative to the KF end-points, with which KF processor error estimates are synchronized.

Consider a single, synchronous, raw measurement-difference** Y_i, taken interior to the KF cycle. The time $\Delta t_{F,i}$ in Figure 4 then represents the fundamental asynchronism between the measurement of actual processor state x_i at time t_i, and the time t_F for which the filter computes its estimate of the actual processor state x_F.

Mathematically, the relationship between the measurement Y_i and the true processor error state at t_F, x_F, is simply:***

$$Y_i = M_{i,F}\, x_F + v_i \tag{19}$$

where v_i is measurement noise and $M_{i,F}$ is the observation matrix; i.e., the linear relationship between the measurement Y_i and the actual state x_F which the filter is estimating.**** $M_{i,F}$ can in particular be written:

$$M_{i,F} = M_i\, \phi_{i,F} \tag{20}$$

where M_i is the linear relationship between Y_i and x_i (the true state at time t_i, which Y_i directly measures) and $\phi_{i,F}$, the (backward) error state transition matrix from time t_F to time t_i.

To synchronize the measurement Y_i to the error state <u>estimate</u> for time t_F, $\hat{x}_F$, the processor KF measurement-residual is therefore formulated as $(Y_i - M_{i,F}\, \hat{x}_F)$, to compensate the effects of actual error state transition in the interval $\Delta t_{F,i}$. Failure to do this [i.e., use of just $(Y_i - M_i\, \hat{x}_F)$], can lead to serious errors in KF operation.

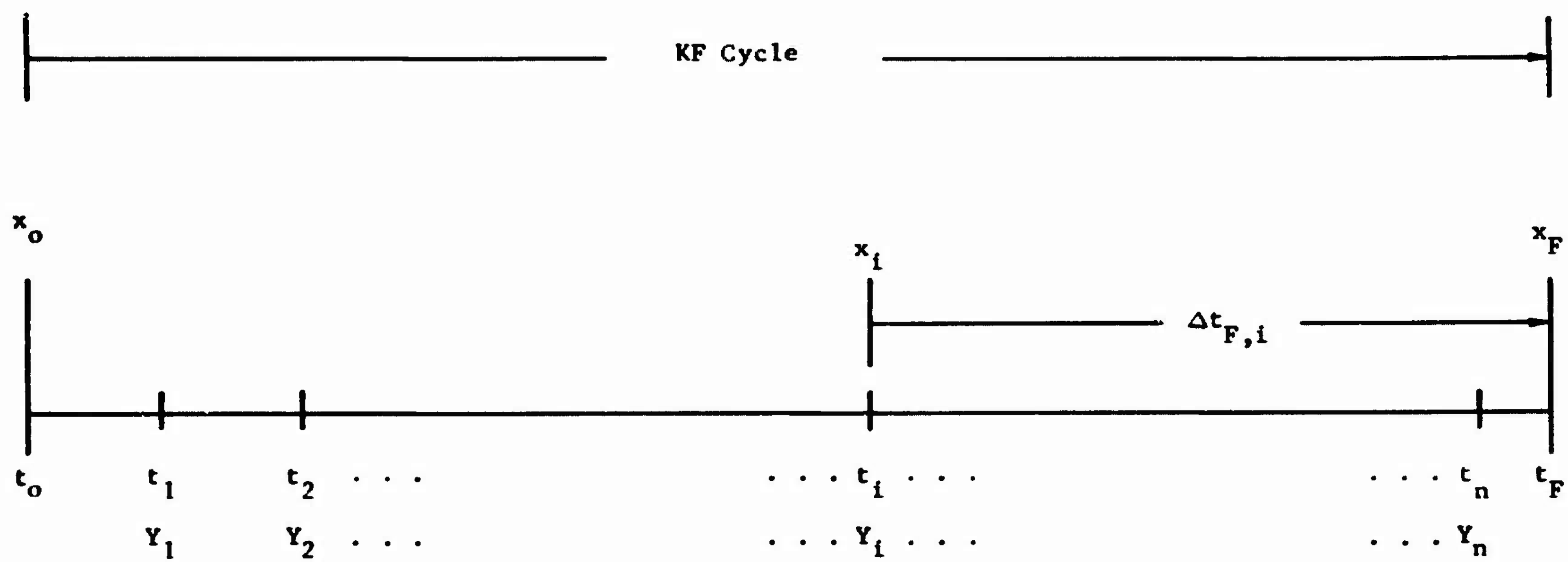
---

*See also Appendix XII.

**It is emphasized here again that synchronous, D/R measurement differencing essentially removes vehicle dynamics, leaving only the dynamics of processor <u>error</u> in its overall estimates (e.g., VSTM position and velocity) of vehicle dynamics.

***System error state noise and control have purposely been neglected here to simplify the discussion.

****See Appendix VI.



t_o, t_F:	KF cycle endpoints (KF processor error estimate synchronization times)
x_o, x_F:	Actual processor error state at times t_o and t_F
t_i:	Times of KF raw measurement availability
Y_i:	Available KF raw measurements
$\Delta t_{F,i}$:	Asynchronism (delay) between Y_i and x_F

Figure 4. Kalman Measurement/Estimate Timing Diagram

For example, suppose Y_i is a two-way range measurement, so that, neglecting propagation errors, Y_i is essentially a measurement of the component of processor position error along the line of sight (and the position error partition of M_i is in fact just the transposed unit LOS vector). If a significant uncompensated velocity error exists along the LOS at the time of measurement, then use of the uncompensated residual formulation above is tantamount to neglecting that part of the position error at t_F due to propagation of the velocity error at time t_i in the interval, $\Delta t_{F,i}$. The accuracy of both KF position and velocity error estimation could therefore be significantly affected.

The compensation actually applied is more general than implied by this example, taking additionally into account (1) any control rates applied to the estimator (e.g., those modeling the leveling control rates actually applied to the IMU in IDR) in the $\Delta t_{F,i}$ interval, since these can, independently of other effects, produce significant velocity errors, and (2) vehicle dynamics-dependent velocity errors generated by vehicle maneuvers in the $\Delta t_{F,i}$ interval (e.g., velocity errors produced in IDR by platform/computer misalignments during aircraft maneuvers).

(b) Measurement Time Smoothing (KMMM)*

If the number of measurements Y_i (for the moment, assumed here to be of a single type; e.g., all LOS pseudoranging measurements on a single emitter) in a single KF cycle is large, then time smoothing techniques -- to an extent and in a form governed by many factors -- can be applied.

These techniques, like the measurement combination techniques discussed in paragraph (c), are desirable to the extent that they reduce the number of measurements which must be processed by the time-consuming KFIM operations (i.e., the actual Kalman filtering of the measurements) into a smaller, prefiltered set of measurements. When they are used, accuracy is thus traded off to gain execution time.

Perhaps the simplest approach is to straightforwardly average the data Y_i; i.e., use the average:

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i \tag{21}$$

as a Kalman filter measurement. Generalizing on the results of paragraph (a), however, such a measurement should be compensated by use of the observation matrix:

$$\bar{M}_F = \frac{1}{n} \sum_{i=1}^{n} M_i \, \phi_{i,F} \tag{22}$$

*See Appendices VII and VIII.

Generation of this matrix is simplified (1) if the data is equispaced; (2) if the M_i can be treated in common as a constant matrix (e.g., if the emitter is a distant navigation satellite, so that its LOS is nearly fixed), and (3) if the pertinent submatrices of $\phi_{i,F}$ depend on only slowly varying dynamics (e.g., the vehicle is not maneuvering). Under these special circumstances, $\bar{M}_F$ can be computed in a single, closed-form computation. If significant data rate irregularity, LOS directional change, and/or vehicle dynamics are present, however, $\bar{M}_F$ must be computed recursively, which requires a computation corresponding to each measurement in the average. The extent to which executing these relatively more time-consuming recursive computations is feasible depends on, among other things, computer speed and processor accuracy requirements for the particular application being considered.

If the measurement data rate is sufficiently high, either formulation for M_F (recursive or single-pass) can be simplified by condensing (i.e., preaveraging) the measurements over short time intervals onto the interval centerpoints and using these as a basis for M_F generation. For example, if the measurement data were such as to much more densely cover the KF cycle interval shown in Figure 4, it might be profitably condensed into n measurements $Y_1, Y_2, \ldots, Y_n$, each consisting of the local average of the actual measurements centered on $t_1, t_2, \ldots, t_n$ and $\bar{M}_F$ could be computed as if there were only n measurements. The viability of this technique is of course dependent on the adequate plus-and-minus cancellation of local velocity-into-position error propagation effects over the short averaging intervals.

(c) KF Measurement Combination (KMCM)

To this point, the discussion has focused on preprocessing KF measurements of a single type. This paragraph deals with the precombination of different types of raw (or KF endpoint-synchronized or time smoothed) KF measurement differences before KFIM use. Two principal types of measurement combination are discussed here: linear and nonlinear. Of these, several generally applicable linear techniques are discussed, the purpose of each of which is principally -- like the linear time-smoothing techniques discussed above -- to reduce the computational load on the computer at some (as yet unknown) cost in accuracy. On the other hand, only a single, more or less special-purpose, but quite important nonlinear combination algorithm is discussed, whose purpose is to enable use by the Kalman filter of LOS net pseudoranging data in the face of large LOS directional uncertainties.

- Linear Techniques: Two specific techniques -- space averaging, and the familiar hyperbolic differencing -- are discussed here; there is also a final, brief general discussion of other linear techniques.

  - Space Averaging:* One of the problems often associated with use of a Kalman filter to accomplish statistically optimum radio psuedoranging, is the need to carry a large -- and therefore computationally costly -- propagation delay error model in the filter. This technique can prove valuable in such situations by eliminating the need for such a model.

    Given a set of either actually simultaneous multichannel-derived or computationally synchronized** single-channel-derived range measurement differences from a number of different emitters, this technique involves simply first converting each of these into a corresponding LOS error vector, vectorially averaging*** all of these vectors, and using the resulting, average error vector as a KFIM (single 3x1, or three 1x1) measurement.****

    The underlying rationale here is simply that since the emitter directions will tend to be uniformly distributed, the error in the averaged measurement will be significantly reduced. For example, if all the n emitters are producing data of the same statistical quality, then the standard deviation of the radial error associated with their average is $1/\sqrt{n}$ times that associated with any one of the vectors separately.

  - Hyperbolic Differencing: This well-known technique, viewed from the standpoint of the more general pseudoranging process (of which it is a special case), implemented by the processor, is another special-purpose, linear measurement combination technique.

    This technique, which essentially consists of linearly differencing pseudoranging measurements in pairs, thereby eliminates emitter net-receiver clock phase difference from the resulting measurements. There is therefore also no need to carry a storage- and time-consuming clock error model in the Kalman filter.

---

*See Appendix VII.

**E.g., KF endpoint-synchronized, using the technique described earlier.

***This can be a simply weighted average if emitters of several different types are involved.

****The KFIM observation matrix used to process this averaged measurement is a simple function of the LOS directions and the endpoint-synchronizing observation matrices associated with all of the measurement differences.

Although there is no geometrical loss in positioning accuracy resulting from hyperbolic differencing of two or more pseudoranging measurements (as opposed to separately processing them)*, there is a loss in both accuracy and operational capability, in that single-emitter data cannot be used (unless a clock error model is carried). This latter fact therefore represents the central tradeoff between the two approaches.

- Other Linear Combination Techniques: In the KMCM module, provision is made for assigning weights to the pseudoranging measurements to be combined in a general way, which is not restricted solely to those special weights associated with the space averaging and hyperbolic techniques discussed above. This has been purposely done to allow for future inclusion of other linear combination techniques.

- Nonlinear Technique:** An important processor design problem in the LOS pseudoranging area centers on the proper use of LOS net pseudoranging measurements in prospective operational situations where the user/emitter relative (3-d) position uncertainties are comparable in size to the actual user-emitter ranges themselves. This produces large LOS directional uncertainties which preclude KFIM use of the normally formulated pseudoranging measurement differences in conjunction with their attendant, normally formulated observation matrices, since the linear measurement-state relationships inherently assumed in the latter no longer hold.

  Two common operational situations in which this problem can occur are (1) navigation start-up with LOS data when little or no information about vehicle position with respect to the net is known, and (2) switching from globally referenced (e.g., NAVSAT-aided) navigation to locally referenced (e.g., target area LOS net-aided) navigation, when, although the vehicle position is accurately known in the E frame, because the local emitter net tie-in to the E frame might be coarse, vehicle position relative to this net is again uncertain.

  Northrop has discovered*** a very attractive technique for solution of this problem, which allows Kalman filter estimation of vehicle position no matter how large (but of course within the computational range of the computer employed) the relative vehicle-emitter net position uncertainty.

*There is a common misconception that the hyperbolic technique suffers uniquely from "gdop" problems near emitter net baselines and their extensions. The fact is that the rho-rho and the pseudoranging techniques suffer identically in this regard -- i.e., all three techniques are equally geometry-sensitive.

**See Appendix IX.

***May 1972. First published in the Multilateration Phase II Status Report No. 1, Period 11 May to 10 June 1972.

The technique simply involves using simple, nonlinear combinations of the pseudoranging measurements in pairs (instead of the individual linear, separate, emitter measurements one by one) as direct KFIM input measurements. These nonlinear pair combinations are such as to produce an <u>exact</u> measurement-state relationship (i.e., observation matrix) and thus remove the difficulty.

This important and promising technique -- like the statistically optimal measurement selection technique discussed below -- in particular merit further investigation, development, and evaluation.

(d) Optimal Measurement Selection (KMOM)*

When several different types of radio position fixing, ranging, range-rating, and altitude data are simultaneously available, appropriate sequencing and/or selection logic must be incorporated in the processor to afford best use of the data in the real time available. Such logic can range all the way from that required simply to implement the operator's selections at all times to that necessary to automatically select and/or sequence on a more sophisticated basis, subject at most to occasional operator overrides. One design approach to the latter philosophy, which has the advantage that is is naturally compatible with the Kalman estimation theory and algorithms, is as follows. A unique feature of any Kalman (or modified Kalman) navigational filter, and one which might be put to advantage in designing the measurement selection logic of the processor, consists in the incorporation of its own error statistics. That is, an error covariance matrix which embodies the estimated variances and covariances of the errors in the various navigation error variables being estimated is automatically (and necessarily) carried and updated as an integral part of the overall filter algorithm computations. In particular, one set of computations which is routinely performed whenever a new measurement is used by the filter produces a covariance-change matrix whose diagonal elements represent the error variance decrements which reflect the improved quality of the new estimate resulting from use of the new measurement.

If sets of these variance-decrement elements -- one set for each measurement -- were computed for all of several simultaneously available measurements of different types in advance of their actual use by the filter, these could be used as a basis for several types of rather conceptually attractive, automatic measurement selection algorithms.

For example, suppose that it were desired during a particular phase of a mission, to select measurements so as to minimize vehicle radial position error. Since the variance of the radial position error is simply the sum of the three variance elements in the covariance matrix which correspond to the three components of vehicle position, then it would only be necessary

*See Appendix VII.

to calculate the radial position error variance decrement for each of the available measurements, and select for further processing that measurement which would produce the largest such decrement. This process could, for example, be activated by the operator via the control panel.

As another example, if it were desired during the weapon delivery phase of a mission to select measurements so as to minimize weapon impact miss distance, then an extension of the above technique could be used to accomplish this as follows. The extra information required here would consist of a set of sensitivity coefficients which linearly related the miss distance to the errors in the appropriate vehicle navigational variables being estimated by the Kalman filter. Given these coefficients, which would in general be expressable terms of already computationally available weapon delivery variables, a set of miss distance decrements -- one for each measurement -- could be computed, and that measurement which yielded the largest such decrement selected for further processing from among the candidate measurements. This overall process might also, like that in the above example, be actuated from the control panel as a selectable operation. The essential mathematics underlying this technique is also summarized in Appendix VII. However, a central question which would require resolution in connection with the prospective use of such algorithms revolves around whether or not the time to execute them might not be so long as to make it preferable to omit all sophisticated selection logic, and simply use the time saved in processing as many measurements through the filter as possible in accordance with some simple cyclic rule instead.

(e) Measurement Reasonableness Testing (KMRM)*

Finally, the availability of built-in Kalman filter error statistics also facilitates the incorporation of measurement data reasonableness tests as part of the measurement preprocessing logic. This type of test is in wide use in the industry.

h. Growth Potential

As indicated in subsection 1, the scope of the processor developed to date is limited to (1) basic navigation output (3-d position, 3-d velocity, and vehicle-to-computer frame angular transformation and angular rate) generation only, and (2) processing of input data from IMU, AHRU, CADS, and radio transceiver pseudoranging equipments only.

However, the flexibility and generalizability which have been carefully built into the processor structure will enable easy expansion of its capabilities to also include (a) processing of input data from additional navigation sensor types, such as doppler radar, the wide class of angle-measuring or angulation (as opposed to pseudorange measuring, or ranging) devices, and (b) processing for operations closely related to navigation, such as steering, guidance, weapon delivery, ILS, etc.

*See Appendix VII.

As an example of processor navigation processing adaptability, consider the software approach to use of a strapdown doppler radar, instead of CADS TAS and a wind estimate, in conjunction with AHRU attitude data to accomplish DDR instead of ADR.*

This can be visualized as a two-step process. First, the formulation of a doppler data processing module which would include all operations necessary to convert raw doppler input signal data into error-compensated frequency shift, and finally into 3-d, airframe-referenced, groundspeed vector data. The functions of this module, like those of the processor DR modules specified to date, would be organized into functionally separate and distinct groups to allow processing at different rates (or even complete omission of certain functions), and formulated in vector/matrix terms (which should be widely applicable to the essentially vector-measurement, doppler process) to facilitate common subroutining with other modules. The computations of this module in DDR would essentially replace those of the WASM in ADR.

Second, the DSWM and KSWM (D portion only) modules would require minor modification to accommodate DDR/PDR switching instead of the (highly similar) ADR/PDR. Thus essentially only two already available processor modules would require modification -- and those only slight modification using already developed techniques discussed in this document -- without otherwise disturbing the main body of processor modules.

An an example of extension of processor techniques to other, non-navigation but closely related avionics functions, consider the generation of, say, weapon delivery computations. In general, such computations have the following suggestive characteristics: (1) their principal outputs -- time and/or distance and/or velocity to go before weapon release or launch -- are straightforwardly formulable in terms of vector-matrix, locally referenced computations, and (2) these computations usually require three-dimensional position, velocity, attitude (and sometimes attitude rate) of carrier aircraft, as well as three-dimensional position and velocity of the target and in the same reference frame).

Overall, therefore, weapon delivery computations could be formulated using not only the same vector matrix techniques as already developed for processor navigation, but navigation processor outputs as principal (carrier vehicle data) inputs as well. Such a formulation would of course not only be highly compatible with the navigation processor formulation developed to date, but would share its intramodule flexibilities and other advantages as well.

---

*Assuming for simplicity in this discussion, a navigation hardware complement not including an IMU, but only an AHRU.

SECTION III

PROCESSOR MLI SPECIFICATION

This section presents the navigation processor specification at the machine-and-language-independent (MLI) level of definition. Overall, this section is organized into five main subsections, largely paralleling the organization of the modular processor itself. These are (1) an initial description of the overall processor modular structure, organization, and information flow, followed by the separate descriptions and actual MLI specifications for each of the principal, processor modular groups: (2) the DR (D) navigation modules, (3) the reference navigation measurement (R) modules, (4) the Kalman filter (K) modules, and (5) the initialization and switching modules.

Each module, whatever its module group, adheres to a more or less standard MLI specification format. This consists of a brief introductory description of the specific functional role of the module with regard to overall processor operations, followed by a formal specification consisting of one or more of each of the following, depending on the modular group to which the module belongs: an operations summary table, an input/output summary table, an operations flow diagram, a logic flow diagram, and a data flow diagram. Further, each module-subsection is intended to be as nearly self-contained as possible from a programmer's point of view; i.e., given only certain minimal additional machine, language, and application-specific information (e.g., relative frequency of execution of the algorithms comprising the module), he could actually program the module from this specification. Consistent with this, module and module group-specifying subsections in this section have been arranged as separate, pull-out packages, starting with a right-hand page, module title page, and ending with a left-hand page (which is blank where necessary to create a pull-out package).

Further, the submodular organization of the specification for each module, although largely standardized, still allows a wide flexibility in such important areas as alternate-algorithm substitution, order and relative frequency of algorithm execution, and so forth.

Finally, the depth of module algorithm definition varies with the module and algorithm involved, in accordance with the degree to which single, obviously preferable candidate algorithms can or cannot be specified short of further machine, language, and application-specific information. For example, processing of IDR specific force into C-frame velocity and position vectors in the VSTM can be accomplished essentially in only one way and is so specified; on the other hand, the vector gravity computation, which can be accomplished in several ways, is left as a more general, input/output identification-level specification only. Correspondingly loose algorithm specifications are also used in particular in several of the Kalman filter modules (e.g., KTMM, KMMM) where both closed-form and recursive formulations are available for the same algorithm, the selection depending on the application.

1. OVERALL PROCESSOR ORGANIZATION SUMMARY

Figure 5 is the overall logic data, control marker flow, and module identification diagram for the navigation processor. In particular, the diagram centers on the dynamic navigation execution loop which includes (a) the sequential processing of the three main navigation update module groups -- i.e., the DR navigation (D) modules, the reference navigation measurement (R) modules, and the Kalman filter (K) modules; (b) the switching modules necessary to control the use of these module groups and their computational reference frame --- i.e., the DR navigation mode switching module, the R configuration switching module, the Kalman filter switching module, and the C frame switching module -- and (c) the platform-to-computer coarse alignment module. In addition, the loop includes provision for peripheral execution of any of a variety of possible navigation output computations (e.g., conversion of basic navigation processor outputs into output display coordinates which are not required to maintain processor basic navigation but use its outputs as inputs). Finally, there are two modules outside the dynamic navigation loop whose sole purpose is initial startup of basic navigation.

Entry into each module on each overall navigation loop execution cycle is governed by a marker-controlled, module use decision. These markers may in general be set (actually or conceptually) by any or all of the following means: (a) operator control, (b) hardware-controlled computer interrupts and/or predetermined (i.e., initial input-controlled) relative frequency of execution data, and (c) changes in DR navigation mode, reference navigation measurement configuration, and platform-to-computer alignment status, controlled by the DSWM and RSWM modules.

Of these, full operator control should probably be associated with the use of certain modules (e.g., CSWM, since the operator can best decide just when processor navigation with respect to a local objective area emitter net should begin), while solely automatic control should perhaps be exerted over certain others (e.g., the Kalman filter modules, since the computer can far more quickly assimilate and use the large amount of relatively cryptic, statistically based data generated by the filter). However, the whole area of operator versus automatic control is of course highly dependent on the processor application and its particular man-machine interface philosophy. Complete processor flexibility in this regard has therefore been retained by the simple device of allowing for the capability of operator intervention of any specified degree in switching modules.

Whatever the degree of operator versus automatic control embedded in the DSWM, RSWM, KSWM, and CSWM logic, it is assumed that the first two produce as respective outputs, on every main loop cycle, the basic DRMM (DR mode marker) and RMCMs (reference navigation measurement processing configuration markers), which (a) fully control the use and internal configuration of the D and R modules respectively, and (b) via the KSWM, control the use of only the corresponding appropriate D, R, and D/R Kalman filter estimate vector

1

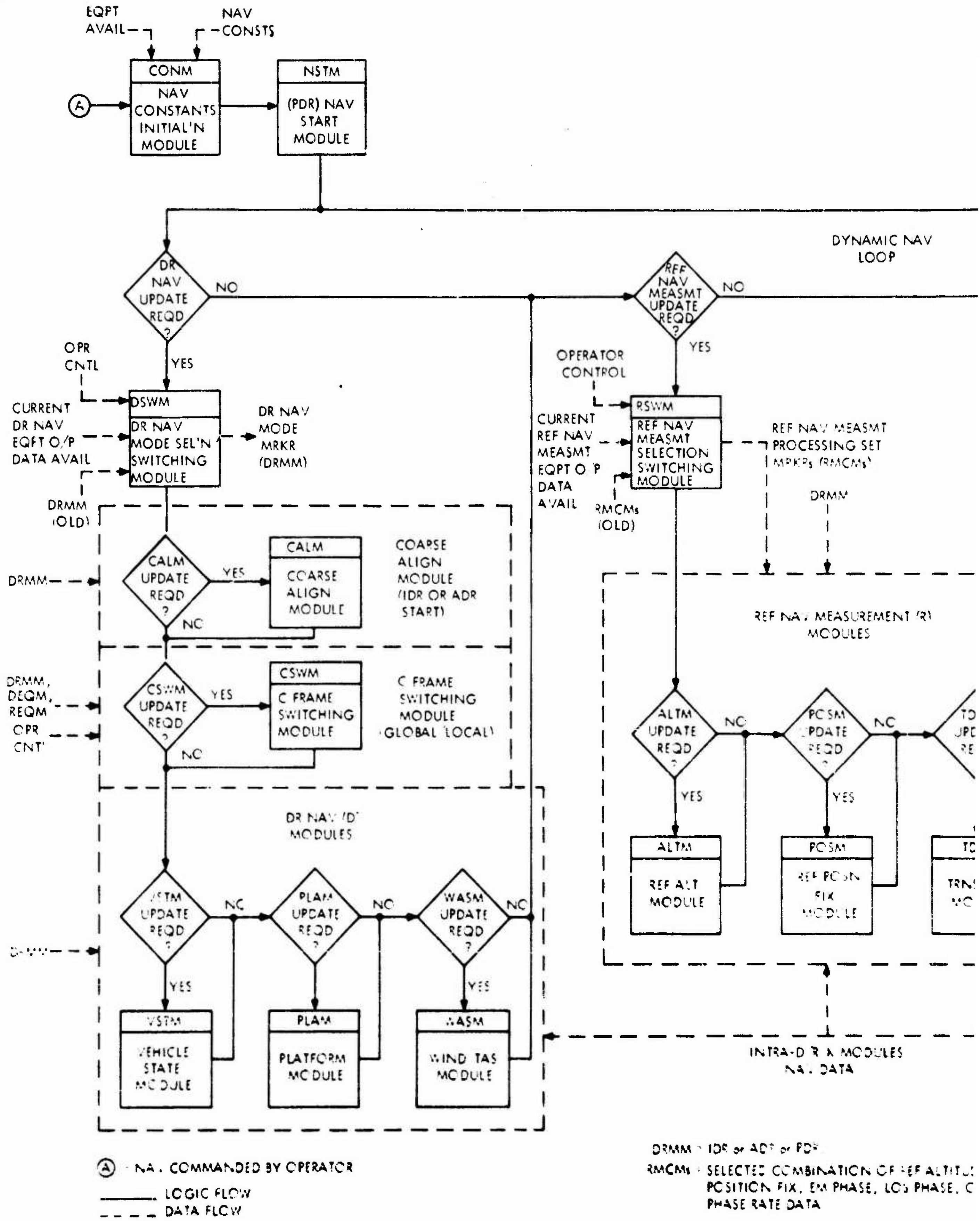
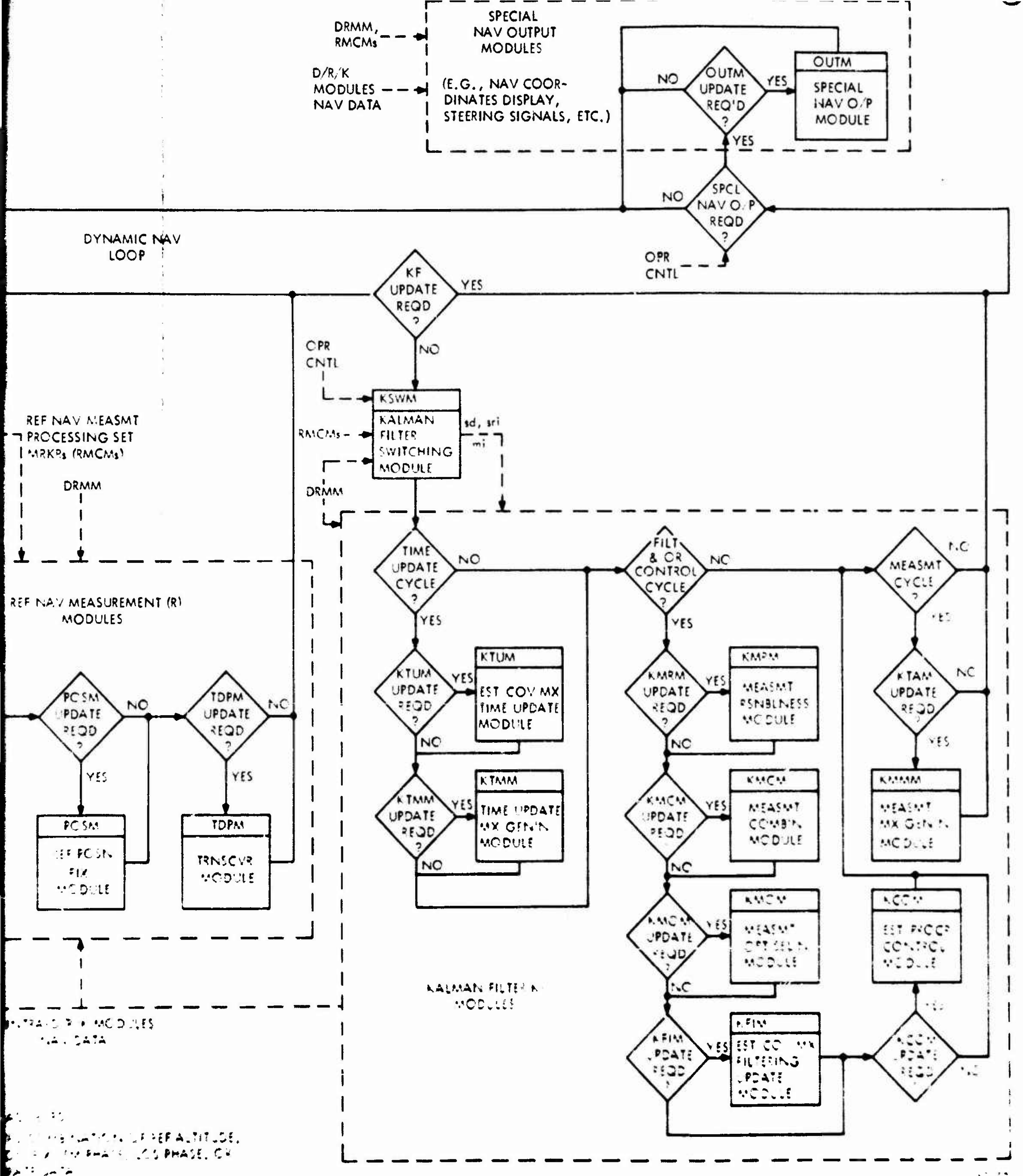


Figure 5. Overall Navigation Processor Flow Diagram


DRMM, RMCMs
D/R/K MODULES NAV DATA
SPECIAL NAV OUTPUT MODULES
(E.G., NAV COORDINATES DISPLAY, STEERING SIGNALS, ETC.)
OUTM UPDATE REQ'D ?
NO
YES
OUTM
SPECIAL NAV O/P MODULE
SPCL NAV O/P REQD ?
OPR CNTL
DYNAMIC NAV LOOP
KF UPDATE REQD ?
OPR CNTL
KSWM
KALMAN FILTER SWITCHING MODULE
RMCMs
DRMM
sd, sri
mi
REF NAV MEASMT PROCESSING SET MRKPs (RMCMs)
REF NAV MEASUREMENT (R) MODULES
PCSM UPDATE REQD ?
PCSM
TDPM UPDATE REQD ?
TDPM
TRNSCVR MODULE
TIME UPDATE CYCLE ?
KTUM UPDATE REQD ?
KTUM
EST COV MX TIME UPDATE MODULE
KTMM UPDATE REQD ?
KTMM
TIME UPDATE MX GEN'N MODULE
FILT & OR CONTROL CYCLE ?
KMRM UPDATE REQD ?
KMRM
MEASMT RSNBLNESS MODULE
KMCM UPDATE REQD ?
KMCM
MEASMT COMB'N MODULE
MEASMT OPT SEL'N MODULE
EST COV MX FILTERING UPDATE MODULE
MEASMT CYCLE ?
KTAM UPDATE REQD ?
MEASMT MX GEN'N MODULE
EST PROCP CONTROL MODULE
KALMAN FILTER (K) MODULES

and covariance matrix partitions by each of the K modules. The KSWM further controls use and internal configuration of the K modules as well. Finally, the CSWM controls the configuration of the (limited) C-frame-dependent portion of the VSTM module. All these switching modules, in addition to the control functions just described, also execute the required navigation variable switching attending each D mode and/or R configuration change for its pertinent module group.

Specification of an appropriate set of predetermined, inter- and intra-module relative frequency of algorithm execution data is correspondingly left entirely open, since this depends heavily on the carrier vehicle dynamical capabilities, mission accuracy requirements, computer speed, and so forth, associated with the specific processor application contemplated.

A summary description of the functions performed by each of the processor modules is included below as a compact, overall-processor, companion reference to Figure 5.

a. Start-Up Modules

(1) CONM: Navigation Constants Initialization Module: Checks for availability of, and installs, all navigation constants required by every processor module for the given navigation equipment configuration.

(2) NSTM: Navigation Start Module: Initializes C frame (C=E), and initializes VSTM and KFMD substates for PDR start.

b. Switching Modules

(1) DSWM: DR Nav Mode Selection/Switching Module: Selects current DR nav mode, based on current DR nav equipment output data availability and operator control. Initializes and switches DR module and coarse align module variables and operations as required by DR mode change.

(2) RSWM: Ref Nav Measurement Selection/Switching Module: Selects set of measurement types for R module and KF module time update) processing, based on current reference navigation measurement equipment, output data availability, and operator control. Initializes and switches R module variables and operations as required by reference navigation processing set change.

(3) KSWM: Kalman Filter Switching Module: Initializes and switches Kalman filter module D, R, and D/R substates and operations, as required by changes in DR nav mode or ref nav measurement processing configuration or computational reference (C) frame.

(4) CSWM: C Frame Switching Module (Non-KF Modules): Initializes and switches D and R module variables (or substates) and operations as required by C frame change (operator controlled).

c. CALM: Coarse Align Module: Coarse initializes (IMU or AHRU) platform-to-computer transformation and local level-to-computer transformations, and coarse levels IMU (as required) prior to IDR or ADR processor operation.

d. DR Nav Modules

(1) VSTM: Vehicle State Module: Continuously updates vehicle state (position and velocity) in C frame.

(2) PLAM: Platform Module: Continuously computes all platform-use-related quantities necessary to provide (a) C frame acceleration inputs to VSTM and control rates to platform control loops in IDR, or (b) VSTM C frame velocity, in conjunction with WASM operation, in ADR.

(3) WASM: Wind/TAS Modules: Continuously computes (a) C frame wind and TAS vectors for VSTM velocity determination in ADR, and (b) C frame TAS vector for wind determination in IDR.

e. Ref Nav Measurement Modules

(1) ALTM: Ref Altitude Module: Continuously computes reference altitude, based on RSWM-selected reference altitude mode, and synchronously differences with VSTM-computed altitude.

(2) POSM: Position Fix Module: Converts panel-entered visual position fix data from input to internal (C frame) coordinates, and synchronously differences the result with VSTM position.

(3) TDPMs: Transceiver Data Processing Modules: Acquisition- and rate-aids radio navigation (ranging) signal processing (TAAM), defines and processes emitter data word signal into emitter position, velocity and antenna lever arm data (TEWM), computes appropriate antenna lever arm corrections for the TRRM module (TALM), computes synchronous VSTM/emitter ephemeris data-based range and range rate for TMOM use (TRRM), computes signal propagation error corrections for the TMOM module (TPCM), computes D/R radio measurement-difference observables (TMOM), computes the measurement matrix (TMMM) and measurement data statistics (TDSM) for KF use in conjunction with the TMOM measurement-differences.

f. Kalman Filter Modules

(1) KTUM: KF Estimate/Covariance Matrix Time Update Module: Time updates KF estimate and attendant covariance matrix over last KF cycle.

(2) KTMM: KF Time Update Matrix Generation Module: Generates current-cycle time update matrices for use by KTUM in next KF cycle.

(3) KMRM: KF Measurement Reasonableness Module: Tests last-cycle measurements for reasonableness. Rejects unreasonable measurements for further KF processing.

(4) KMCM: KF Measurement Combination Module: Linearly combines some or all of the set of reasonable, last-cycle measurements (i.e., the set passed by the KMRM) and its set of attendant measurement matrices.

(5) KMOM: KF Measurement Optimal Selection Module: Optimally orders the set of last-cycle linearly combined measurements for use by the KFIM.

(6) KFIM: KF Estimate/Covariance Matrix Filtering Module: Updates the KF processor error estimate and its attendant covariance matrix, using the set of last-cycle, reasonable, and linearly combined measurements (and their attendant measurement matrices) which are outputted by the KMRM, KMCM, and KMOM modules. Resulting estimate and covariance matrices are synchronized with start of current cycle.

(7) KMMM: KF Synchronized Measurement Matrix Generation Module: Generates time-smoothed, estimate-synchronized measurement matrices for use with current cycle measurements in next KF cycle.

(8) KCOM: KF Estimate/Processor Control Module: Generates and executes end-of-cycle KF estimate and non-KF module variable corrections and controls.

g. Special Navigation Output Modules

This is a class of prospective, special-purpose modules, which use nav processor outputs as principal inputs (e.g., output display coordinate computation module, steering signal generation module, etc.)

III.2

DR NAVIGATION (D) MODULES

SPECIFICATIONS SUMMARY

This module group, which executes basic DR navigation, consists of the central, all-mode (IDR, ADR, PDR) Vehicle State Module (VSTM), augmented by the Platform Module (PLAM) and the Wind/Airspeed Module (WASM) in IDR or ADR.

The specification for each of these three modules is composed of (a) a brief summary description of the module functions, (b) a DR mode-dependent, module operations summary table, (c) a DR mode-controlled, module operations flow diagram, and (d) a DR mode-dependent, module input/output summary table. In particular, use is made throughout each specification of the following DR mode-use set mnemonics, for identifying and grouping these operations themselves, as well as their inputs and outputs:

I,P,A =	IDR, PDR, ADR modes, respectively	
IPA =	Set of Operations (or input/output)	common to I, P, and A modes
IP =	" "	common to I and P modes
IA =	" "	common to I and A modes
PA =	" "	common to P and A modes
I =	" "	exclusive to I mode
P =	" "	exclusive to P mode
A =	" "	exclusive to A mode

Thus, for example, in the IDR mode, as the operations flow diagrams prescribe, all of the (non-null) operation sets IPA, IP, IA, and I, as identified in the operations summary table, need to be executed, and the corresponding input/output requirements are summarized against these sets in the input/output summary table.

The relative frequency of execution of each of the separate operations in the summary table is purposely left unspecified, since it is highly application-dependent. On the other hand, the order has been carefully selected such that, if followed on the first module execution cycle in any new DR mode, it leads to minimum DSWM navigation variable switching requirements. If another order is used, therefore, the DSWM logic and operations must be carefully reexamined and revised as necessary.

III.2.a

VEHICLE STATE MODULE

(VSTM)

SPECIFICATION

This module operates in all three DR modes.

In IDR, operation consists of resolution of PLAM-generated specific force into the C frame, its compensation for gravity and Coriolis accelerations, and its integration into C-frame-referenced velocity and position vectors.

In ADR, the C-frame-referenced velocity vector is computed as the sum of the WASM-derived airspeed and wind vectors, and integrated into a C-frame-referenced position vector.

In PDR, an L-frame-referenced pseudoacceleration vector is resolved into the C frame and integrated into C-frame-referenced velocity and position vectors. The pseudoacceleration vector is also appropriately time-decayed.

In particular, the C-frame-referenced vehicle position, velocity, and gravity vectors, and the geoidal altitude, are generated in all three DR modes.

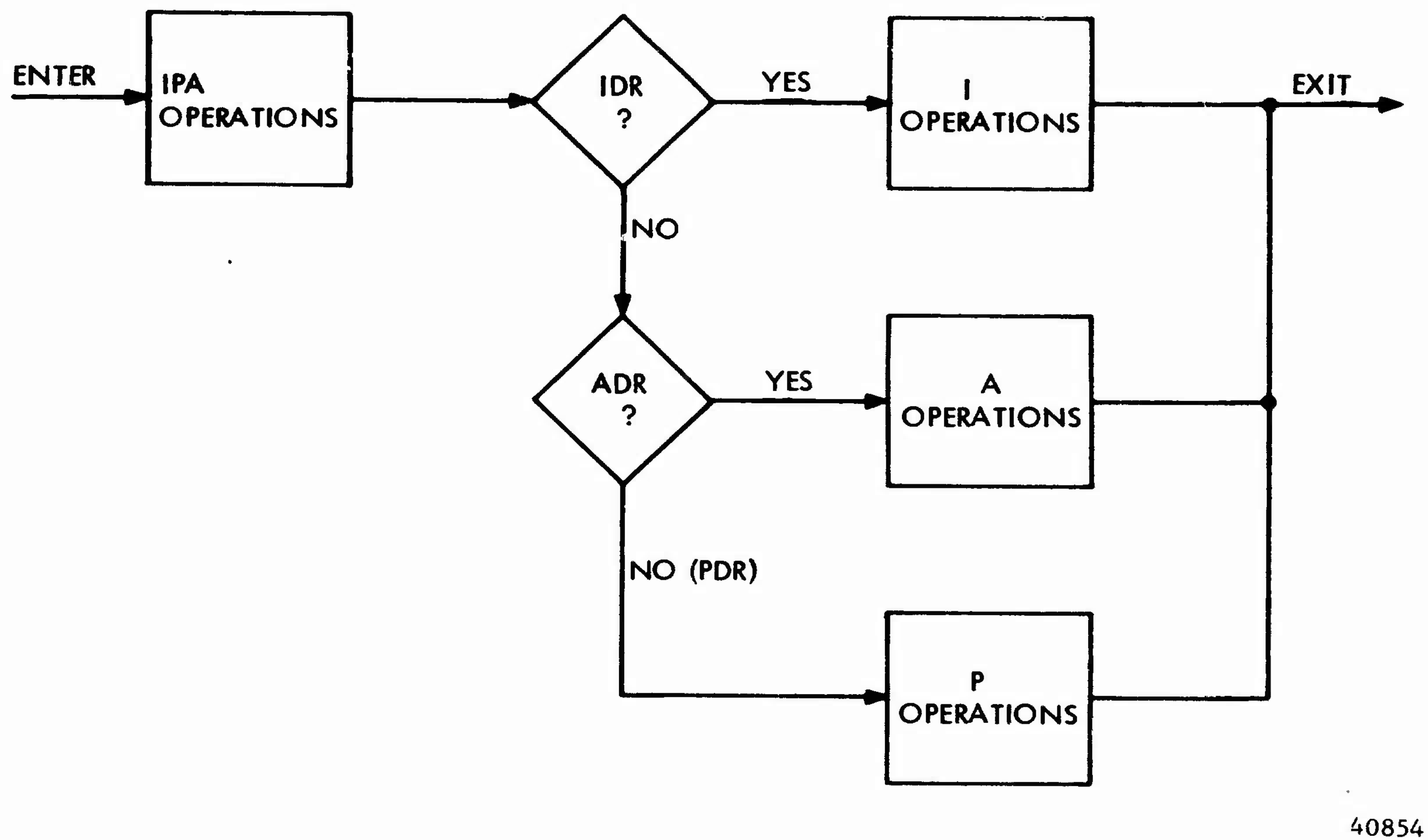


Figure 6. VSTM Operations Flow

TABLE V. VSTM OPERATIONS SUMMARY

<table>
<tr><th colspan="2">DR Nav Mode / Operation</th><th>IDR(I)</th><th>ADR(A)</th><th>PDR(P)</th></tr>
<tr><td rowspan="10">Intra-KF Cycle Execution</td><td>C Frame Referenced Acceleration (in L Frame)</td><td>---------</td><td>--------</td><td>$|a_L| = |v|^2 - (v^T u_1)^2$
$(u_1)^T_L = [1\ 0\ 0]$ (P1)
$(\dot{v}_C)_L = \beta_L - |a_L|(u_1)_L$</td></tr>
<tr><td>L/C Frame Transf., $T_{L/C}$</td><td>---------</td><td>----------</td><td>$u_1 = \left|\frac{g}{g}\right|$, $\gamma = \left|\frac{v}{v}\right|$,
$u_2 = \frac{\gamma \times u_1}{|\gamma \times u_1|}$, $u_3 = u_1 \times u_2$ (P2)
$T_{L/C} = [u_1\ u_2\ u_3]^T$</td></tr>
<tr><td>Specific Force, f</td><td>$f = T^T_{P/C} f_P$ (I1)</td><td>---------</td><td>---------</td></tr>
<tr><td>Velocity, v</td><td>$\Delta v = \int_{\Delta t_v} (f + g - 2\omega_{E/I} \times v)\, dt$ (I2)
$v^k = v + \Delta v$</td><td>$v^k = v_W + v_{AS}$ (A1)</td><td>$v^k = v + T^T_{L/C} \int_{\Delta t_v} (\dot{v}_C)_L\, dt$ (P3)</td></tr>
<tr><td>Position, p</td><td colspan="3">$\Delta p = \int_{\Delta t_P} v\, dt$, $p^k = p + \Delta p$ (IPA1)</td></tr>
<tr><td>Position, p_E</td><td colspan="3">$p_E = T^T_{C/E}\, p + C_{C/E}$ (IPA2)</td></tr>
<tr><td>Gravity, g_E</td><td colspan="3">$g_E = g_E(p_E)$ (IPA3)</td></tr>
<tr><td>Gravity, g</td><td colspan="3">$g = T_{C/E}\, g_E$ (IPA4)</td></tr>
<tr><td>Geoidal Altitude h</td><td colspan="3">$h = |p_E| - |p_S(p_E)|$ (IPA5)</td></tr>
<tr><td>Maneuver Acceleration, β_L (L Frame)</td><td>----------</td><td>--------</td><td>$Q^a_{\beta L} = e^{-k_{\beta L}\Delta t_a}$ (P4)
$\beta^k_L = Q_{\beta L}\beta_L$</td></tr>
<tr><td colspan="5">k: Variables Controlled by KCOM at KF Cycle Endpoints
a: $e^{-k_{\beta L}\Delta t_a}$ = Diagonal Matrix With Terms $e^{-k_{\beta Li}\Delta t_a}$; $k_{\beta Li} = \frac{1}{\tau_{\beta Li}}$;
$\tau_{\beta Li}$ = Constant, or Controlled by Flight Control Data</td></tr>
</table>

40891

TABLE VI. VSTM INPUT/OUTPUT SUMMARY

Inputs/Outputs	DR Mode Subsets	IPA	I	A	P
Inputs	Dynamic	$T^*_{C/E}$, $C_{C/E}^*$, Δt_p	f_P, $T^*_{P/C}$, Δt_v, $\omega_{E/I}$	v^*_W, v^*_{AS}	Δt_v, Δt_a
	Constant				
Outputs	Dynamic	v^*; p^*; Δp^*; p_E, g_E, g^*; h, p_S	f, Δv^*		$(v_C)_L$, β_L, $\lvert a_L \rvert$, u_1^*; u_2^*; u_3, $T^*_{L/C}$, $Q_{\beta L}$
	Constant				

* C Frame Dependent

III.2.b

PLATFORM MODULE

(PLAM)

SPECIFICATION

This module operates only in the IDR or ADR modes. In addition, IDR operation is further broken down into strapdown IMU-only (IS) and rotationally isolated (free) IMU-only (IF) operations, and ADR operations into those which are required only when the AHRU is in the DG mode (AD), and only when it is in the magnetic mode (AM).

Whatever the DR mode, IMU type, or AHRU mode, the PLAM always generates the following principal outputs: (a) the basic interframe transformation matrices and relative angular rate vectors between the platform, locally level, and (earth-fixed) computer frames, and (b) (if appropriate platform readout data is available) the corresponding entities relating the vehicle (=air) and computer frames.

In addition in IDR, the PLAM error-compensates IMU accelerometer outputs, shapes and outputs the platform frame-referenced specific force vector, and error-compensates and shapes IMU gyro control signals.

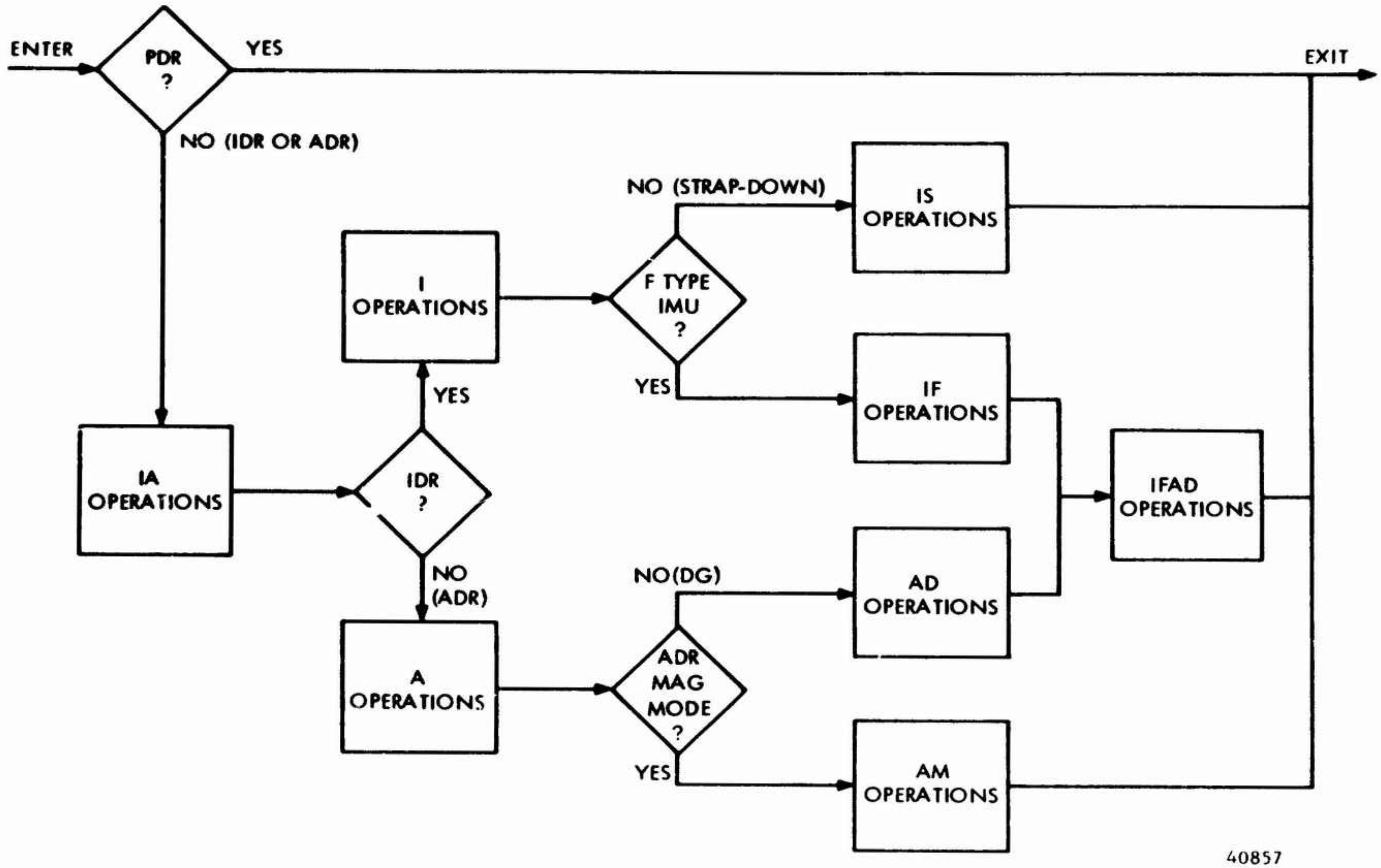


Figure 7. PLAM Operations Flow

TABLE VII. PLAM OPERATIONS SUMMARY

Operation \ DR Nav Mode	IDR(I) Strapdown (IS)	IDR(I) Rot. Free (IF)	ADR(A) (DG Mode (AD)	ADR(A) Mag Mode (AM)
Intra-KF Cycle Execution				
Earth Rate $(\omega_{E/I})^L$ (I Frame)	$(\omega_{E/I})_L = T_{L/C}\omega_{E/I}$ (IA1)			
Geoidal Curvature Matrix K_L	K_L = Function of $h, (\omega_{E/I})_L / \lvert\omega_{E/I}\rvert$ (IA2)			
L/C Frame Ang. Rate $(\omega_{L/C})_L$	$(\omega_{L/C})_L = K_L T_L \; v$ (IA3)			
L/C Frame Transf. $T_{L/C}$	$T_{L/C}^{k} = T_{L/C} - \int_{\Delta t_{L/C}} (\omega_{L/C})_L \times T_{L/C}\,dt$ (IA4)			
P/I Frame Ang. Rate $(\omega_{P/I})_P$ (IS)	$(\omega_{P/I})_P = \omega_{GYR} - \Delta\omega - \Delta\omega_K^{k}$ (IS1)	----------		
P/L Frame Ang. Rate $(\omega_{P/L})_P$	$(\omega_{P/L})_P = (\omega_{P/I})_P - T_{P/L}\{(\omega_{L/C})_L + (\omega_{E/I})_L\}$ (IS2)	$(\omega_{P/L})_P = \omega_K^{k}$ (IFAD1)		a
P/L Frame Transf. $T_{P/L}$	$T_{P/L}^{k} = T_{P/L} - \int_{\Delta t_{L/C}} (\omega_{P/L})^P \times T_{P/L}\,dt$ (IA5)			
P/C Frame Ang. Rate $(\omega_{P/C})_P$	$(\omega_{P/C})_P = (\omega_{P/L})_P + T_{P/L}(\omega_{L/C})_L$ (IA6)			
P/C Frame Transf. $T_{P/C}$	$T_{P/C} = T_{P/L}T_{L/C}$ (IA7)			
Platform Specific Force Calibr'n Δf	Δf = Funct. of $f_P, (\omega_{P/I})_P$, PACC (IS3)	Δf = Funct. of $f_P, (\omega_{P/I})_P$, PACC (IF1)	----------	
Platform Inertl. Ang. Rate Calibr'n $\Delta\omega$	$\Delta\omega$ = Funct. of $f_P, (\omega_{P/I})_P$, PGCC (IS4)	$\Delta\omega$ = Funct. of $f_P, (\omega_{P/I})_P$, PGCC (IF2)	----------	
Platform Specific Force f_P	$f_P = f_{ACC} + \Delta f + \Delta f_K^{k}$ (II)		----------	
P/I Frame Ang. Rate $(\omega_{P/I})_P$ (IF)	----------	$(\omega_{P/I})_P = (\omega_{P/C})_P + T_{P/L}(\omega_{P/I})_L + \Delta\omega_K^{k}$ (IF3)	----------	
Gyro Torquing Rate ω_{GYR}	(ω_{GYR} = Strapdown Gyro Sensed Rate)	$\omega_{GYR} = (\omega_{P/I})_P + \Delta\omega$ (IF4)	----------	
A/P Frame Tranxf. $T_{A/P}$	($T_{A/P}$ = Const. Matrix)	$T_{A/P}$ = Funct. of IMU Att. Readoutx (IF5)	$T_{A/P}$ = Funct. of AHRU Att. Readouts [illegible]	
A/P Frame Ang. Rate $(\omega_{A/P})_P$	($(\omega_{A/P})_P = 0$)	$(\omega_{A/P})_P$ = Funct. of IMU Att. & Att. Rate Readouts (IF6)	$(\omega_{A/P})_P$ = Funct. of AHRU Att. & Att. Rate Readouts [illegible]	
A/C Frame Trans. $T_{A/C}$	$T_{A/C} = T_{A/P}T_{P/C}$ [illegible]			
A/C Frame Ang. Rate $\omega_{A/C}$	$\omega_{A/C} = T_{P/C}^{T}\{(\omega_{A/P})_P + (\omega_{P/C})_P\}$ [illegible]			

** Variables Controlled by KCOM at KF Cycle Endpoints

$$* \quad (\omega_{P/L})_P = \left[\frac{(v^T e')(g^T \omega_{E/I})}{g\, P_E^T\, e' \omega_{E/I}}\right] T_{P/C}\left(\frac{g}{|g|}\right) + \omega_K^{k} \qquad (e' = \omega_{E/I} \times g)$$

TABLE VIII. PLAM INPUT/OUTPUT SUMMARY

Inputs/Outputs	DR Mode Subsets	IA	I	A	IF	IS	AM	IFAD
Inputs	Dynamic	$h, v^*, \omega^*_{E/I}, \Delta r_{L/C}$	$f_{ACC}, \Delta f_k, \Delta\omega_k$	AHRU Att., Att.Rate Readouts	IMU Att., Att. Rate Readouts	ω_{GYR}	$g^*, P_E, \lvert g \rvert$	ω_K
	Constant	$\lvert \omega_{E/I} \rvert$	PACC, PGCC					
Outputs	Dynamic	$(\omega_{E/I})_L, (\omega_{L/C})_L, (\omega_{P/L})_P, (\omega^*_{P/C})_P, (\omega_{A/P})_P, \omega^*_{A/C}, T^*_{L/C}, T_{P/L}, T^*_{P/C}, T_{A/P}, T^*_{A/C}, K_L$	$(\omega_{P/I})_P, \Delta f, \Delta\omega$		ω_{GYR}			
	Constant							

*C Frame Dependent

III.2.c

WIND/AIRSPEED MODULE

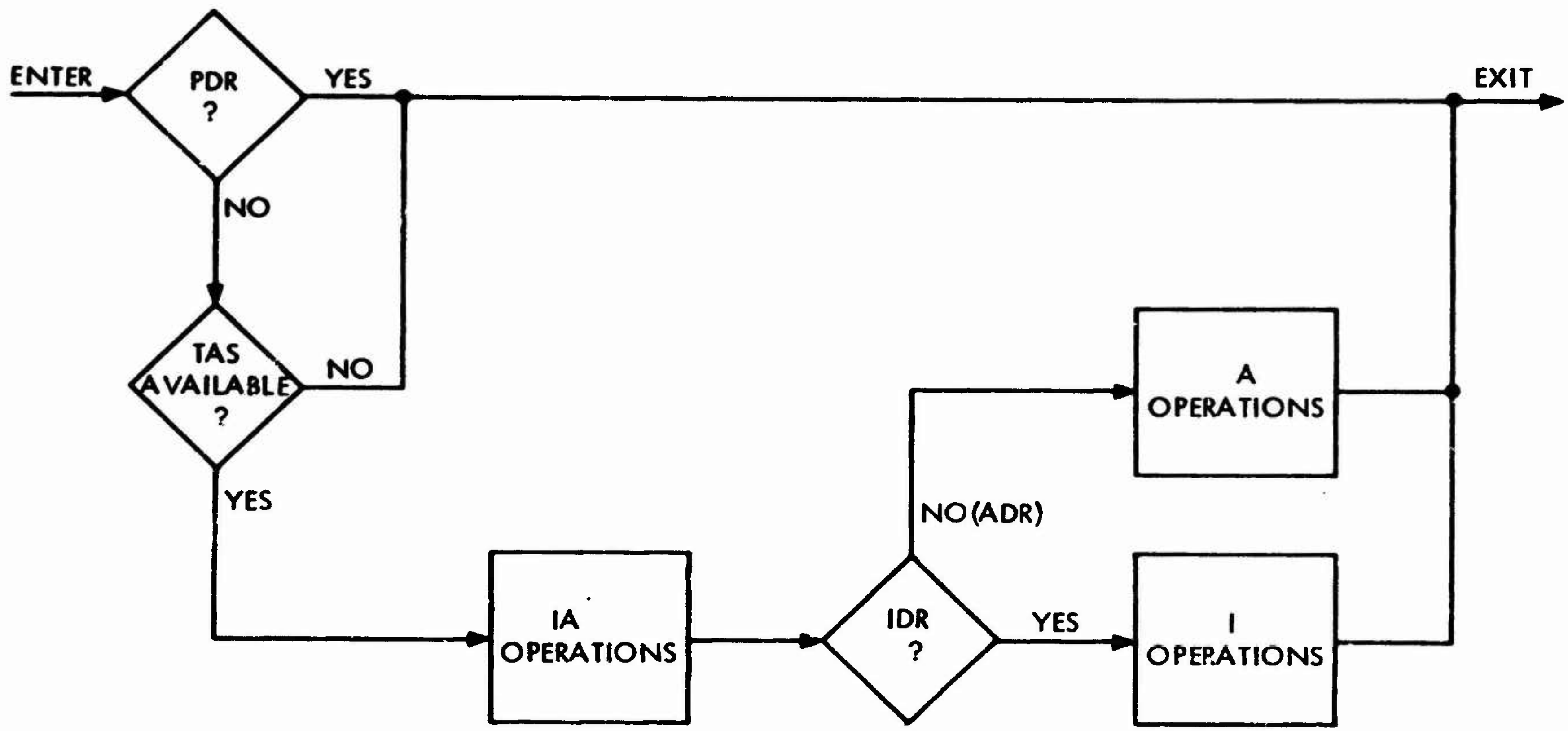
(WASM)

SPECIFICATION

This module operates only in the IDR or ADR modes.

In the ADR mode, operations involve error-compensation, shaping, and C frame resolution of CADS true airspeed and L frame wind estimates, into C frame airspeed and wind vector outputs. In addition, the wind estimate is appropriately time-decayed.

In the IDR mode, the process is reversed, and an L-frame-referenced wind estimate is continuously determined from the C-frame-referenced airspeed and groundspeed vector difference. In particular, the groundspeed vector used is just the VSTM velocity vector.



40855

Figure 8. WASM Operations Flow

TABLE IX. WASM OPERATIONS SUMMARY

Operation \ DR Nav Mode		IDR(I)*	ADR(A)	PDR(P)
Intra-KF Cycle Execution	TAS Calibration Δa	Δa = Function of a_m (IA1)		---------------
	Compensated TAS a	$a = a_m + \Delta a + \Delta a_K^k$ (IA2)		---------------
	TAS Angle of Attack Compensation	$T_{A/A'}, k_{A/A'}$ = Functions of Angle of Attack (IA3)		---------------
	TAS Vector $(v_{AS})_A$ (A Frame)	$(v_{AS})_A = T_{A/A'} k_{A/A'} a$ (IA4)		---------------
	TAS Vector v_{AS} (C Frame)	$v_{AS} = T_{A/C}^T (v_{AS})_A$ (IA5)		---------------
	Wind Vector v_W (C Frame)	$v_W = v - v_{AS}$ (I1)	$v_W = T_{L/C}^T (v_W)_L$ (A2)	---------------
	Wind Vector $(v_W)_L$ (L Frame)	$(v_W)_L = T_{L/C} v_W$ (I2)	$(v_W)_L^k = Q_{WL}^{**} (v_W)_L$ (A1)	---------------
KF Cycle Endpoint Execution				

40982

*With TAS available. This module is not used in IDR if TAS is not available.

k: Variables controlled by KCOM at KF cycle endpoints.

**$Q_{WL} = e^{-K_{WL}\Delta t_W}$ = Diagonal Matrix with Terms $e^{-k_{WLi}\Delta t_W}$; $k_{WLi} = \frac{1}{\tau_{WLi}}$

TABLE X. WASM INPUT/OUTPUT SUMMARY

Inputs/Outputs	DR Mode Subsets	IPA**	IP**	IA**	PA	I**	P	A
Inputs	Dynamic			$T^*_{L/C}, T^*_{A/C}, \Delta a, \Delta a_K$		v^*		
	Constant							Q_{WL}
Outputs	Dynamic			$v^*_W, (v_W)_L, v^*_{AS}, (v_{AS})_A, a, a_m$ $T_{A/A'}, k_{A/A'}$				
	Constant							

*C Frame Dependent

**IDR with TAS available. This module is not used at all if TAS is not available.

III.3

REFERENCE NAVIGATION MEASUREMENT (R) MODULES

SPECIFICATIONS

There are three of these modules, corresponding to the three types of reference navigation measurement data assumed. These are (a) the reference altitude module (ALTM), (b) the position fix module (POSM), and (c) the transceiver data processing module (TDPM).

Unlike the D modules, where a high degree of interrelationship between modules exists, the R modules are essentially independent of one another, and each R module specification is therefore arranged by itself in this subsection as a separate, pullout package.

III.3.a

REFERENCE ALTITUDE MODULE

(ALTM)

SPECIFICATION

Operation of this module depends partly on whether the aircraft is on the ground or in the air. When airborne, the best available reference altitude -- barometric altitude from the CADS or pseudo altitude in its absence -- is continuously selected and updated by this module. This reference altitude is then synchronously differenced with DR altitude computed by the VSTM, for measurement preprocessing use by the Kalman Filter Measurement Matrix Generation Module (KMMM).

On the ground (where it is assumed that panel-inserted field altitude is always available) field altitude is selected as the best available reference altitude when the barometric altitude is not available; when available, barometric altitude itself -- continuously corrected to equal the field altitude -- is selected instead. Kalman filter use of this data does not commence until the vehicle is airborne.

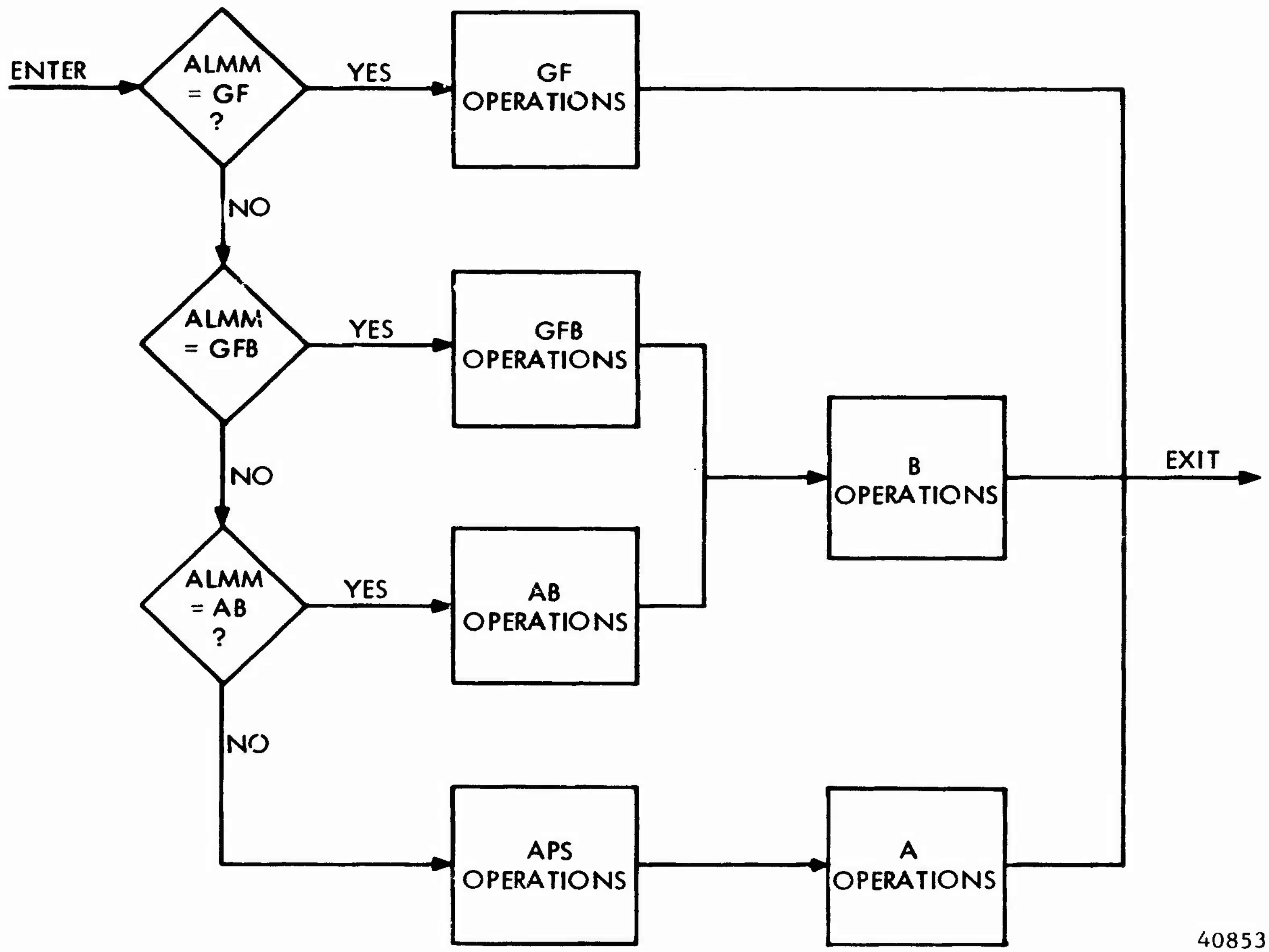


Figure 9. ALTM Operations Flow

TABLE XI. ALTM OPERATIONS SUMMARY

Operation / Ref Altitude Mode		Air(A)		Ground(G)	
		Pseudo Alt (No Baro Avail)	Baro Alt Available	Field Alt Plus Baro Alt	Field Alt Only
		APS	AB	GFB	GF
Intra-KF cycle Execution	Ref Alt h_R	(APS1) $h_R = h_{PC}$	(B1) $h_R = h_{BC}$		(GF1) $h_R = h_F$
	Corrected Baro Alt h_{BC}		(AB1) $h_{BC} = h_B + \Delta h_{BK}$	(GFB1) $h_{BC} = h_F$	
	Corrected Pseudo Alt h_{PC} Update	(APS2) $h_{PC} = h_{PC} + k_h(h_{PC} - h_{CR})$	----------------		(GF2) $h_{PC} = h_F$
	Δh_{BK} Corr'n (Ground)	------------------------		(GFB2) $\Delta h_{BK} = h_F - h_B$	
	Synchronous Alt Diff Δh	$\Delta h = h - h_R$		--------------	

Note: $\Delta h_{BK}, h_{PC}$ Corrected by KF in Airborne Modes only.

TABLE XII. ALTM INPUT/OUTPUT SUMMARY

Alt Mode ⟶	APS	AB	GFB	B	G	BG	APS/GF	A
Input	h_{CR}, k_h	Δh_{BK}	----	h_B	h_F	-----	----	h
Output	----	----	Δh_{BK}	h_{BC}	----	h_R	h_{PC}	Δh

III.3.b

REFERENCE POSITION FIX MODULE

(POSM)

SPECIFICATION

This module converts panel-inserted visual position fix data from input coordinates to an internal C frame-referenced position vector, and synchronously differences this vector with the DR position vector generated by the VSTM.

TABLE XIII. POSM OPERATIONS SUMMARY

Intra-KF-Cycle Execution	Ref Position Vector p_R Generation	p_R = Function of Panel Input Position Fix Coordinates
	Synchronous Position Diff. Δp	$\Delta p = p - p_R$

TABLE XIV. POSM INPUT/OUTPUT SUMMARY

Input	p, Panel Input Position Fix Coordinates
Output	p_R, Δp

III.3.c

TRANSCEIVER DATA PROCESSING MODULES

(TDPM)

SPECIFICATIONS SUMMARY

The fundamental TDPM submodule organization is shown in Figure 10. The module interface to the receiver is defined by the two inputs which consist of the received data word message and the basic range and range rate signals measured by the receiver. An acquisition and aiding signal is provided to the receiver interface from the TDPM. Three intramodule outputs are provided by the TDPM: (a) the filter variances provided by the TDSM, (b) the measurement matrix elements generated by the TMMM, and (c) the Kalman filter observable data obtained from the TMOM. Elements of the Kalman filter estimates are provided for correction to several other modules within the TDPM to complete the intramodule interfacing.

Two data bus outputs dominate the submodule interfacing. These are the basic emitter word data vector generated by the TWEM and the scalar range and rate information developed by the TRRM. These two dominant information flows govern the operation of the other six submodules and only two other intermodule interfaces are required to complete the functional architecture.

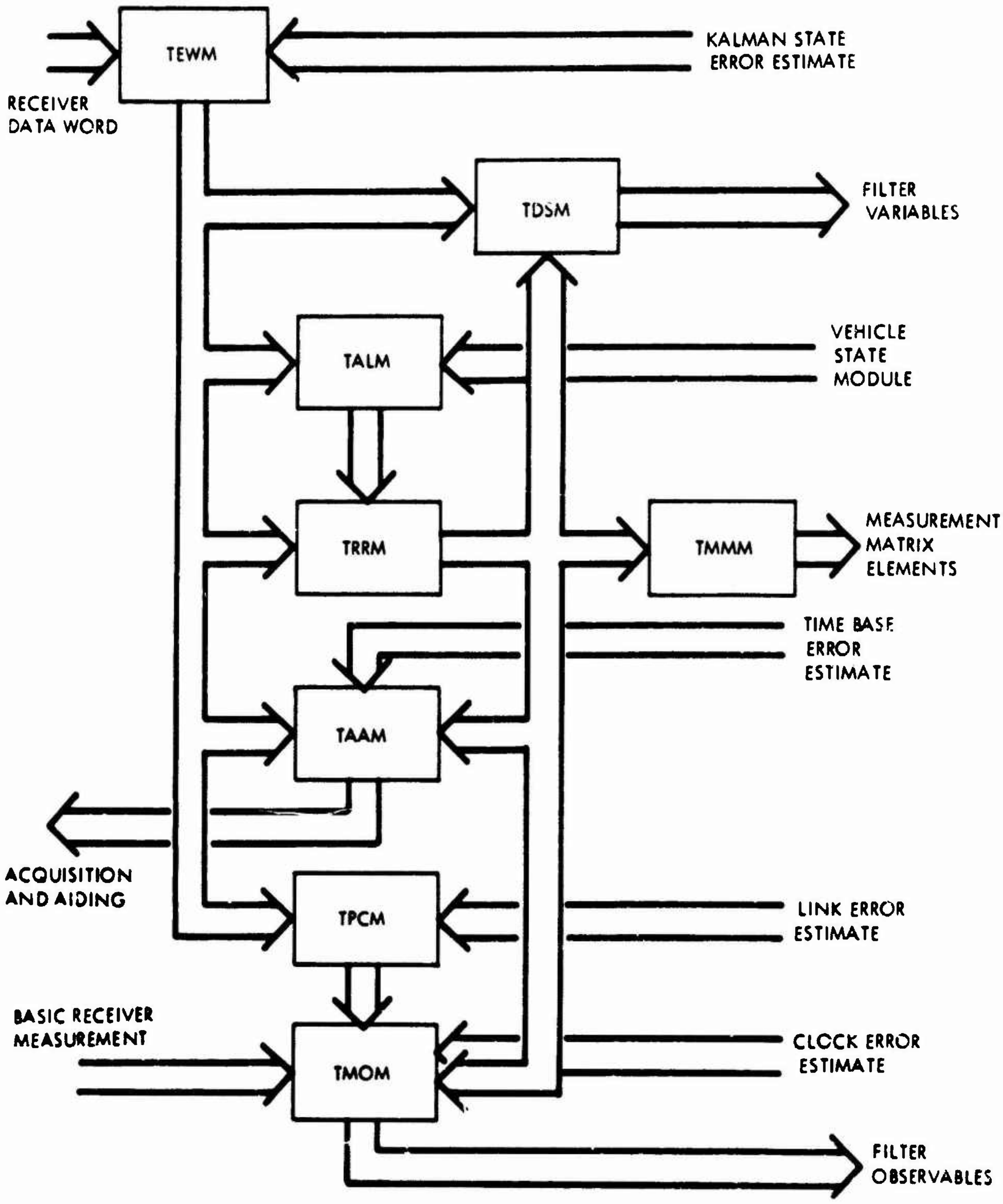


Figure 10. TDPM Modules Organization/Receiver Interface

III.3.c.(1)

TRANSCEIVER ACQUISITION AND AIDING MODULE

(TAAM)

SPECIFICATION

The receiver acquisition and signal tracking processes are enhanced if information is provided defining the initial phase and phase rate and current deviations in this data. For a time division multiplexed system (TDM) the estimated time of arrival of the emitter signal is also indicated to the receiver. Rate aiding in terms of the phase derivatives allows for narrowband tracking within the receiver loops.

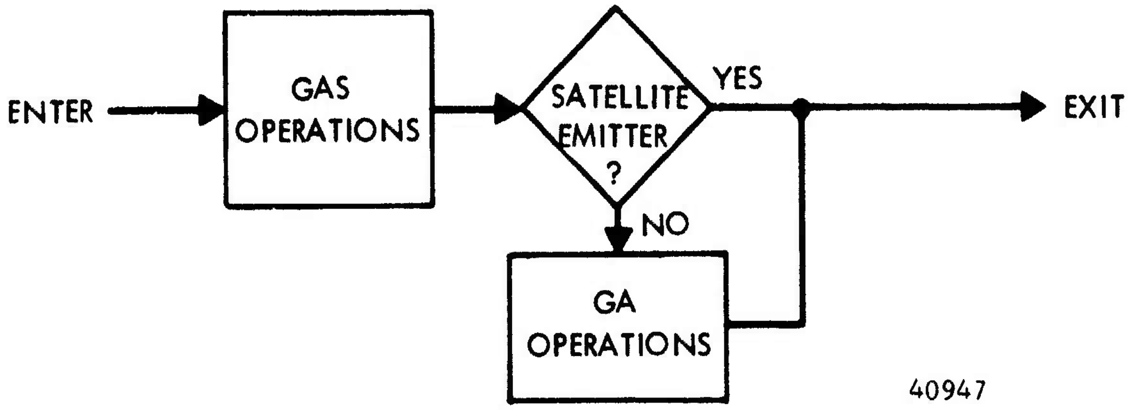


Figure 11. TAAM Operations Flow

TABLE XV. TAAM OPERATIONS SUMMARY

Operations \ Transceiver Configuration		Ground Emitter (G)	Airborne Emitter (A)	Satellite Emitter (S)
Intra-KF-Cycle Execution (Once per Emitter Type j)	Time of Arrival (TDM System) t_j	$t_{Dj} = \Delta_K t_j + \frac{\lvert R_j \rvert}{C}$ (GA1)		Already calculated in TEWM
		$t_j = t_u - t_{Dj}$ (GAS1)		
	Phase ϕ_j Indication (CDM System) and Initialization	$\phi_j = K_\phi\left(\lvert R_j\rvert + \lvert R_j\rvert \Delta T_d + r_j^T \ddot{R}_j \frac{\Delta T_d^2}{2}\right)$ (GAS2)		
	Phase Search Deviation $\Delta\phi$	$\Delta\phi_j = K_\phi\sqrt{r_j^T P_{D11} r_j}$ (GAS3)		
	Rate Aiding $\dot{\phi}_j$	$\dot{\phi}_j = K_{\dot{\phi}}\left(\lvert \dot{R}_j\rvert + r_j^T \ddot{R}_j \Delta T_d\right)$ (GAS4)		
	Frequency Search Deviation $\Delta\dot{\phi}$	$\Delta\dot{\phi}_j = K_{\dot{\phi}}\sqrt{r_j^T P_{D22} r_j}$ (GAS5)		

TABLE XVI. TAAM INPUT/OUTPUT SUMMARY

Input Constants

K_{ϕ} = range-to-phase conversion

$K_{\dot{\phi}}$ = velocity-to-phase-rate conversion

ΔT_d = computational delay constant

Input Variables

t_u = user receiver time base (1x1) } TEWM

$\Delta_K t_j$ = cumulative KF emitter clock error correction (1x1) } TEWM

r_j = unit jth emitter LOS vector (3x1)

$|R_j|$ = estimated scalar range to jth emitter (1x1) } TRRM

$|\dot{R}_j|$ = estimated range rate to jth emitter (1x1) } TRRM

P_{D11} = KF position error covariance matrix (3x3)

P_{D22} = KF velocity error covariance matrix (3x3)

$\ddot{R}_j$ = measured range acceleration vector in computational frame (3x1) (if available)

Output Variables

t_j = time of arrival of jth emitter for TDM systems

ϕ_j = phase initialization for receiver acquisition

$\dot{\phi}_j$ = phase rate initialization for receiver acquisition

$\Delta\phi_j$ = phase search extent for receiver acquisition

$\Delta\dot{\phi}_j$ = phase rate extent for receiver acquisition

III.3.c.(2)

TRANSCEIVER EMITTER WORD MODULE

(TEWM)

SPECIFICATION

The demodulated serial bit data stream which is processed by each emitter receiver is converted to distinct emitter position and velocity states by this module. The other elements of the data word which contain information on emitter dynamics or on propagation constants are simply throughputted to other modules. The basic emitter states are also corrected to maintain the updated value suitable for navigation. The data word may also contain time base data, in terms of total state values or in terms of emitter time offsets. If no data word time base is given, the user receiver time state is based on the user clock with corrections.

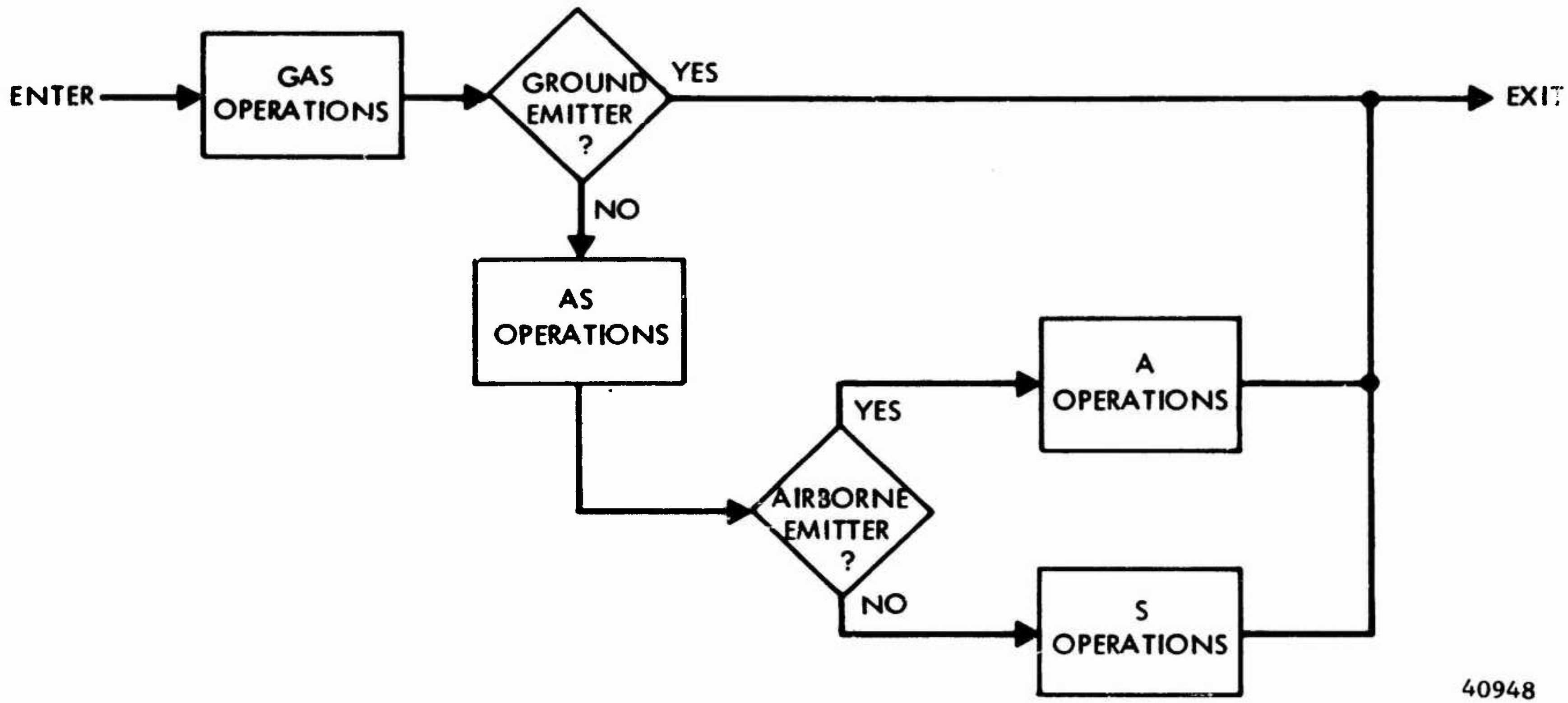


Figure 12. TEWM Operations Flow

TABLE XVII. TEWM DATA WORD INPUTS

Transceiver Configuration / Data Word	Ground Emitter (G)	Airborne Emitter (A)	Satellite Emitter (S)
Time Base		t_j	
Position	Not required	e_j	Not required
Velocity	Not required	$\dot{e}_j$	Not required
Antenna Arm	Not required	$(d_{EMj})_A$	Not required
Direction Cosine	Not required	$T_{C/AEMj}$	Not required
Attitude Rate	Not required	$\omega_{AEMj/C}$	Not required
Surface Refraction	N_s		
Satellite Coefficient	Not required		A_{ji}
Ionosphere Coefficient	Not required		I_{ji}

TABLE XVIII. TEWM OPERATIONS SUMMARY

Operations \ Transceiver Configuration		Ground Emitter (G)	Airborne Emitter (A)	Satellite Emitter (S)
Intra-KF-Cycle Execution (Once per Emitter Type j)	Initialization of Time t'_u	$t'_u = t_j$ (with data word) (GAS1)		
		$t'_u = t_u$ (no data word) (GAS2)		
	Corrected Time t_u	$t_u = t'_u + \Delta_K t_u$ (GAS3)		
	Delay Time t_D	Not required		$t_{Dj} = \Delta_K t_j + \frac{\lvert R_j \rvert}{C}$ (S1)
	Determination of Emitter Position e'_j	$e'_j = e'_j[0]$ $e'[0]$=constant	e'_j = data word (A1)	$t_j = (t_u - t_{Dj})$ (S2) $e'_j = \sum_{i=0}^{n} A_{ji}(t_j)^i$ (S3)
	Determination of Emitter Velocity $\dot{e}'_j$	Not required	$\dot{e}'_j$ = data word (A2)	$\dot{e}'_j = \frac{d}{dt} e'_j$* (S4)
	Corrected Emitter Position e_j*	$e_j = e'_j + \Delta_K e_j$ (GAS4)		
	Corrected Emitter Velocity $\dot{e}_j$	Not required	$\dot{e}_j = \dot{e}'_j + \Delta_K e_j$ (AS1)	

*$\dot{e}'_j$ may be formed by an alternate series of $\dot{e}'_j = \sum_{i=0}^{n} A_{ji}(t)^i$

C = speed of light(constant)

TABLE XIX. TEWM INPUT/OUTPUT SUMMARY

Inputs:

$\Delta_K t_j$ = cumulative KF emitter clock error correction (1x1)

$\Delta_K t_u$ = cumulative KF user receiver clock error correction (1x1)

t_u = user receiver time base (1x1)

$|R_j|$ = estimated scalar range to the jth emitter (1x1)

$\Delta_K e_j$ = cumulative KF emitter position error correction (3x1)

$\Delta_K \dot{e}_j$ = cumulative KF emitter velocity error correction (3x1)

Data Word = demodulated serial bit stream consisting of the following data:

- t_j = emitter time base (1x1)
- e'_j = emitter position vector for jth emitter (3x1)
- $\dot{e}'_j$ = emitter velocity vector for jth emitter (3x1)

$(d_{EMj})_A$ = antenna lever arm (3x1)

$T_{C/AEMj}$ = airframe-to-C-frame transformation matrix (3x1)

$\omega_{AEMj/C}$ = airframe-to-C-frame angular rate (expressed in C frame) (3x1)

N_s = surface refractivity

A_{ji} = satellite position/velocity polynomial coefficient (ix1)

I_{ji} = satellite ionospheric correction polynomial coefficient (ix1)

Outputs:

Data word elements of:

$\left.\begin{matrix} e_j \\ \dot{e}_j \end{matrix}\right\}$ input to TRRM

$\left.\begin{matrix} (d_{EMj})_A \\ T_{C/AEMj} \\ \omega_{AEMj/C} \end{matrix}\right\}$ input to TALM

$\left.\begin{matrix} N_s \\ I_{jn} \end{matrix}\right\}$ input to TPCM

$\left.t_u\right\}$ input to TAAM

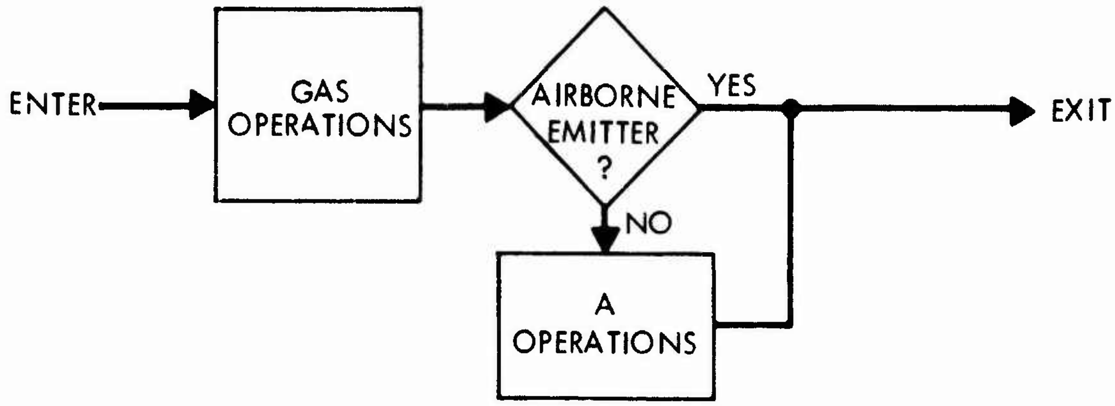
III.3.c.(3)

TRANSCEIVER ANTENNA LEVER ARM MODULE

(TALM)

SPECIFICATION

The compensation for antenna position and angular rate about the computational frame point defined by the platform requires the use of location and attitude rate data. The inclusion of emitter antenna motion for airborne relay transceiver configuration requires additional emitter reporting data which specifies antenna location, attitude, and attitude rate in aircraft coordinates or in a common computational reference frame.



40969

Figure 13. TALM Operations Flow

TABLE XX. TALM OPERATIONS SUMMARY

	Operations \ Transceiver Configuration	Ground Emitter (G)	Airborne Emitter (A)	Satellite Emitter (S)
Intra-KF-Cycle Execution (Once per Emitter Type j)*	User Antenna Displacement d_u	$d_u = T_{C/A} \left(d_u\right)_A$ (GAS1)	$d_u = T_{C/A} \left(d_u\right)_A$ (GAS1)	$d_u = T_{C/A} \left(d_u\right)_A$ (GAS1)
	Emitter Antenna Displacement d_{EMj}	Not required	(A1) $d_{EMj} = T_{C/AEMj} \left(d_{EMj}\right)_A$	Not required
	Antenna User Velocity $\dot{d}_u$	$\dot{d}_u = \omega_{A/C} \times d_u$ (GAS2)	$\dot{d}_u = \omega_{A/C} \times d_u$ (GAS2)	$\dot{d}_u = \omega_{A/C} \times d_u$ (GAS2)
	Emitter Antenna Velocity $\dot{d}_{EMj}$	Not required	(A2) $\dot{d}_{EMj} = \omega_{AEMj/C} \times d_{EMj}$	Not required

TABLE XXI. TALM INPUT/OUTPUT SUMMARY

Inputs

$\left(d_u\right)_A$ = user antenna lever arm in aircraft coordinates (3x1)

$\omega_{A/C}$ = angular attitude rates of user aircraft with respect to computational C frame (3x1)

$T_{A/C}$ = transformation from C frame to airframe (3x3)

$\omega_{AEMj/C}$ = jth emitter airframe-to-C-frame angular rate (3x1)

$\left(d_{EMj}\right)_A$ = jth emitter lever arm in emitter aircraft

$T_{C/AEMj}$ = jth emitter airframe-to-C-frame transformation (3x3)

Outputs (in C frame)

d_u = user antenna lever arm displacement (3x1)

$\dot{d}_u$ = user antenna lever arm displacement rate (3x1)

d_{EMj} = jth emitter antenna lever arm displacement (3x1)

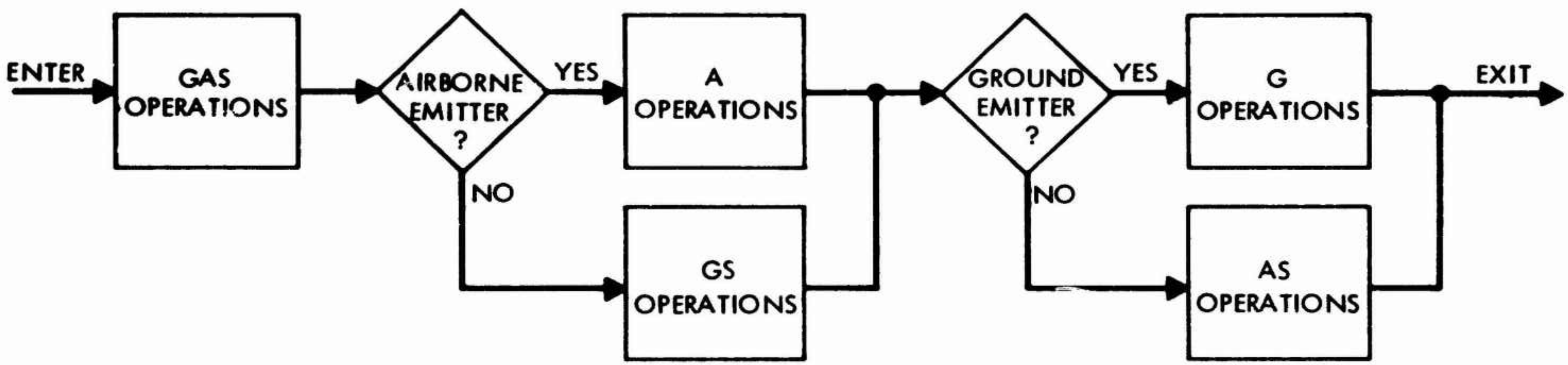
$\dot{d}_{EMj}$ = jth emitter antenna lever arm displacement rate (3x1)

III.3.c.(4)

TRANSCEIVER RANGE AND RANGE RATE MODULE (TRRM)

SPECIFICATION

The determination of the scalar range and range rate to each emitter is estimated from data on the emitter and user vector dynamics. Various compensations for antenna and emitter dynamics are incorporated to provide high accuracy for each emitter type. The vector pointing direction in the form of a unit vector quantity is also determined to provide the coordinate frame orientation for the transceiver measurements.



40971

Figure 14. TRRM Operations Flow

TABLE XXII. TRRM OPERATIONS SUMMARY

Operations \ Transceiver Configuration		Ground Emitter (G)	Airborne Emitter (A)	Satellite Emitter (S)
Intra-KP-Cycle Execution (Once per Emitter Type j)	User/jth Emitter Vector Range R_j	$d = d_u$ (GS1)	$d = d_U - d_{EMj}$ (A1)	$d = d_u$ (GS1)
		$R_j = p + d - e_j$ (GAS1)		
	LOS Range $\lvert R_j \rvert$	$\lvert R_j \rvert = \left(R_j^T R_j\right)^{1/2}$ (GAS2)		
	Unit LOS Vector r_j	$r_j = R_j / \lvert R_j \rvert$ (GAS3)		
	LOS Range Rate $\lvert \dot{R}_j \rvert$	$\dot{d} = \dot{d}_u$ (GS2)	$\dot{d} = \dot{d}_u - \dot{d}_{EMj}$	$\dot{d} = \dot{d}_u$ (GS2)
		$\lvert \dot{R}_j \rvert = r_j^T(\dot{p} + \dot{d})$ (G1)	$\lvert \dot{R}_j \rvert = r_j^T\left(\dot{p} + \dot{d} - \dot{e}_j\right)$ (AS1)	

TABLE XXIII. TRRM INPUT/OUTPUT SUMMARY

Inputs:

$\left.\begin{array}{l} e_j = \text{jth emitter position vector } (3\times1) \\ e_j = \text{jth emitter velocity vector } (3\times1) \end{array}\right\}$ TEWM

$\left.\begin{array}{l} p = \text{estimated receiver position vector } (3\times1) \\ \dot{p} = \text{estimated receiver velocity vector } (3\times1) \end{array}\right\}$ VSTM

$\left.\begin{array}{l} d_u \\ \dot{d}_u \\ d_{EMj} \\ \dot{d}_{EMj} \end{array}\right\}$ TALM Outputs

Outputs:

$|R_j|$ = estimated scalar range to jth emitter (1x1)

$|\dot{R}_j|$ = estimated range rate to jth emitter (1x1)

r_j = unit vector along line-of-sight to jth emitter (3x1)

$\left.\begin{array}{l} (\dot{p}+\dot{d}) = \text{vector velocity to emitter } (3\times1) \\ (\dot{p}+\dot{d}-\dot{e}_j) = \text{vector velocity to emitter } (3\times1) \end{array}\right\}$ Input to TMMM

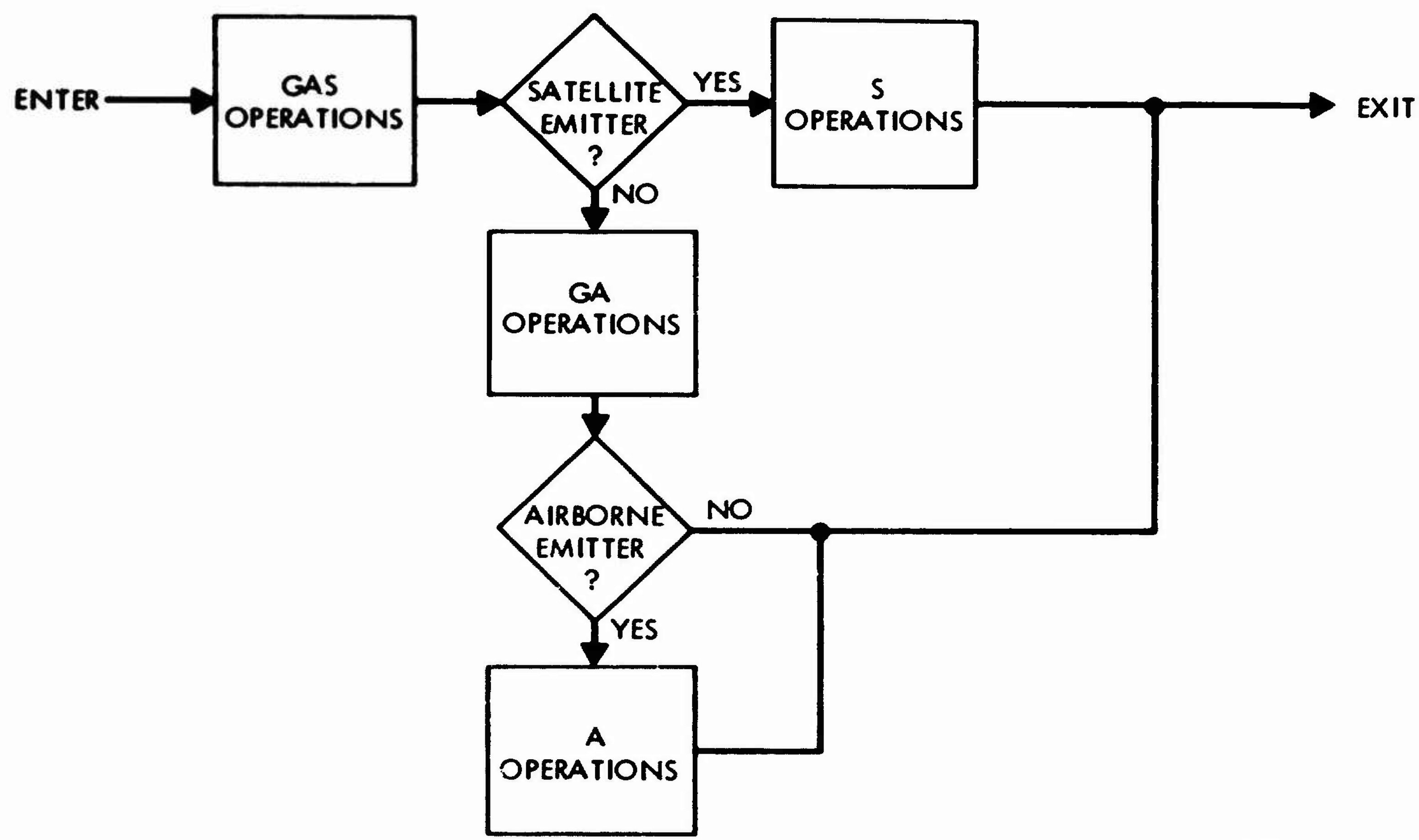
III.3.c.(5)

TRANSCEIVER PROPAGATION CORRECTION MODULE

(TPCM)

SPECIFICATION

The modelable errors for line-of-sight propagation corrections for transceiver configurations are the tropospheric velocity change and ray bending and the ionospheric group delay for satellite emitters. These corrections are based on physical, empirically derived models which employ various parameters as inputs to generate the compensations. Basic emitter-user geometry is also tested to ensure that the most suitable emitter is employed to generate the desired navigation information.



40967

Figure 15. TPCM Operations Flow

TABLE XXIV. TPCM OPERATIONS SUMMARY

	Operations \ Transceiver Configuration	Ground Emitter (G)	Airborne Emitter (A)	Satellite Emitter (S)
Intra-KP-Cycle Execution (Once per Emitter Type j)	LOS Angle β_j	$\beta_j = \frac{\pi}{2} - r_j^T \frac{P_E}{\lvert P_E \rvert}$ (GAS1)		
	Negative Angle Check	Not required	$\lvert p \rvert > \lvert e_j \rvert$ (A1)	Not required
	Redefine LOS Angle β_j	Not required	$\beta_j = -r_j^T \frac{P_E}{\lvert P_E \rvert}$ (A2)	Not required
	Satellite Availability	Not required		$\beta_j < 5°$ (S1)
	Tropospheric Correction $\Delta\rho_{Tj}$	$\Delta\rho_{Tj} = \frac{K_T\ Ns}{(\sin)\beta_j + C_1\ P}$ (GA1)		Additional Algorithm (S2)
	Low Grazing Angle	$\beta_j < 3°$ (GA2)		Not required
	Redefine LOS Angle β_j	$\beta_j = \beta_j - \theta_j$ ** (GA3)		Not required
	Additional Bending Correction	$\Delta\rho_{Tj} = \Delta\rho_{Tj} + K_b CSC\ \beta_j$ (GA4)		Not required
	Ionospheric Correction	Not required		$\Delta\rho_{Ij} = \frac{K_I \hat{I}_v CSC\sqrt{\beta_j^2 + C_2^2}}{f^2}$ (S3)
	Total Correction	$\Delta\rho_{Tj} + \Delta\rho_{Ij} + \Delta_K L_j = \Delta R_{mj}$ (GAS2)		

TABLE XXV. TPCM INPUT/OUTPUT SUMMARY

Inputs:

p = estimated receiver position vector (3x1)

r_j = unit vector along line-of-sight to jth emitter (3x1)

e_j = jth emitter position vector (3x1)

N_s = surface refractivity } TEWM

$\Delta_K L_j$ = cumulative KF jth link propagation error correction (1x1)

$\hat{I}_v$ = electron content (may be calculated from I_{ji} coefficient of TWEM)

Constants:

K_T = tropospheric constant

C_1 = atmospheric constant or functional parameter of altitude

P = exponent constant

θ_j^{**} = bending angle error which may be defined either as a function or as a tabulated parameter of β_j

(*)(**) indicates potential expanded algorithm

K_b = bending constant

K_I = ionospheric constant

f = frequency

C_2 = tropospheric constant or functional parameter

Outputs:

ΔR_{mj} = total propagation correction (to THOM)

III.3.c.(6)

TRANSCEIVER MEASUREMENT OBSERVABLES MODULE (TMOM)

SPECIFICATION

The basic measurement data provided to the Kalman filter estimation algorithm is the difference between the computed and measured values of range and range rate. All known propagation link range and range rate errors must also be corrected before making this comparison. The pseudorange and range rate time offsets due to both emitter and user clock biases are applied as range correction terms.

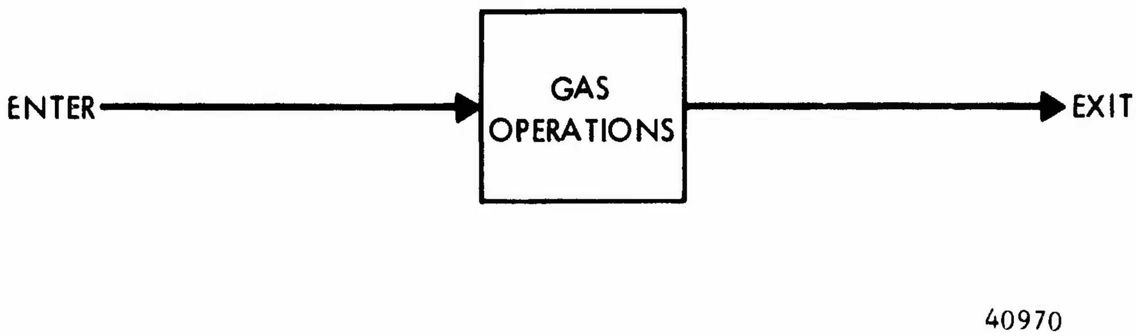


Figure 16. TMOM Operations Flow

TABLE XXVI. TMOM OPERATIONS SUMMARY

<table>
<tr><th colspan="2">Transceiver Configuration / Operations</th><th>Ground Emitter (G)</th><th>Airborne Emitter (A)</th><th>Satellite Emitter (S)</th></tr>
<tr><td rowspan="8">Intra-KF-Cycle Execution (Once per Emitter Type j)</td><td>Measured Range and Range Rate Scaling $R''_{mj}, \dot{R}''_{mj}$</td><td colspan="3">$R''_{mj} = K_1 \phi_j$ (GAS 1)
$\dot{R}''_{mj} = K_2 \dot{\phi}_j$ (GAS2)</td></tr>
<tr><td>Measured Range Propagation Correction R'_{mj}</td><td colspan="3">$R'_{mj} = R''_{mj} + \Delta R_{mj}$ (GAS 3)</td></tr>
<tr><td>Measured Range Rate Propagation Correction $\dot{R}'_{mj}$</td><td colspan="3">$\dot{R}'_{mj} = \dot{R}''_{mj} + \Delta \dot{R}_{mj}$ (GAS4)</td></tr>
<tr><td>Measured Range Clock-Difference Correction R_{mj}</td><td colspan="3">$R_{mj} = R'_{mj} + C\left(\Delta_K t_u - \Delta_K t_j\right)$ (GAS5)</td></tr>
<tr><td>Measured Range Rate Clock Rate-Diff. Correction $\dot{R}_{mj}$</td><td colspan="3">$\dot{R}_{mj} = \dot{R}'_{mj} + C\left(\Delta_K \dot{t}_u - \Delta_K \dot{t}_j\right)$ (GAS6)</td></tr>
<tr><td>Range Difference Observable Y_{Rj}</td><td colspan="3">$Y_{Rj} = \lvert R_j \rvert - R_{mj}$ (GAS7)</td></tr>
<tr><td>Range Rate Difference Observable Y_{RRj}</td><td colspan="3">$Y_{RRj} = \lvert \dot{R}_j \rvert - \dot{R}_{mj}$ (GAS8)</td></tr>
</table>

$K_1 K_2$ = constants for scaling from receiver to computer units

C = speed of light

TABLE XXVII. TMOM INPUT/OUTPUT SUMMARY

Inputs:

Symbol	Description	Source
R_{mj}	= corrected, scaled measured pseudorange from jth receiver channel** (1x1)	
$\dot{R}_{mj}$	= corrected, scaled measured pseudorange rate from the jth channel** (1x1)	
$\lvert R_j \rvert$	= scalar range estimate (1x1)	TRRM
$\lvert \dot{R}_j \rvert$	= range rate estimate (1x1)	TRRM
$\Delta_K t_j$	= cumulative KF emitter clock error correction (1x1)	From KF
$\Delta_K t_u$	= cumulative KF user clock error correction (1x1)	From KF
$\Delta_K \dot{t}_j$	= cumulative KF emitter clock rate error correction (1x1)	From KF
$\Delta_K \dot{t}_u$	= cumulative user clock rate error correction (1x1)	From KF
ΔR_{mj}	= propagation range correction (1x1)	TPCM
ΔR_{mj}	= propagation range rate correction (1x1)	TPCM

Outputs:

Symbol	Description	Destination
Y^*_{Rj}	= observable range error (1x1)	Inputs to KF Measurement Preprocessing
Y^*_{RRj}	= observable range rate error (1x1)	Inputs to KF Measurement Preprocessing

*In the KF Partitioned Structure Specification, the indices R and RR used here and in the TMMM and TDSM modules are respectively denoted simply by 3 and 4

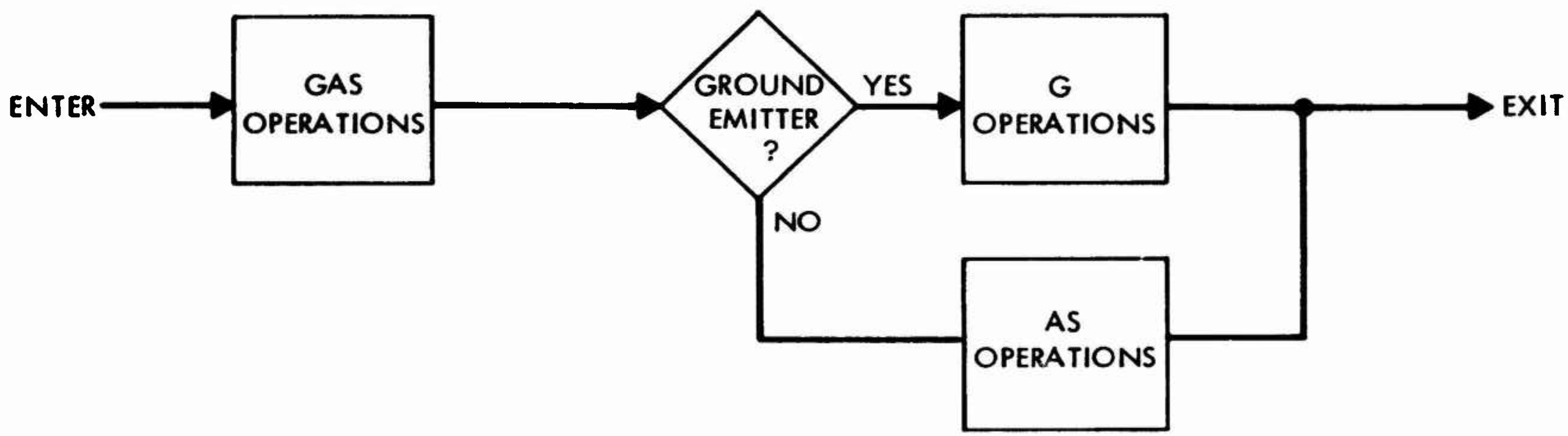
**i.e., from jth emitter via jth receiver channel

III.3.c.(7)

TRANSCEIVER MEASUREMENT MATRIX MODULE (TMMM)

SPECIFICATION

The measurement matrix elements relating the measured observable to the error states are specified by this module. The user and emitter position and velocity errors are determined in terms of the unit LOS (direction cosine) vector. The elements for time and link propagation errors are also established by functional terms, and are related to the range or range rate observations by using the speed of light and by stipulating a single propagation error state.



40968

Figure 17. TMMM Operations Flow

TABLE XXVIII. TMMM OPERATIONS SUMMARY

	Operations \ Transceiver Configuration	Ground Emitter (G)	Airborne Emitter (A)	Satellite Emitter (S)
Intra-KF-Cycle Execution (Once per Emitter Type j)*	User Position Error State	$M_{R\delta p} = r_j^T$ (GAS1)		
	Emitter Position Error State	$M_{R\delta e j} = -r_j^T$ (GAS2)		
	User Time Error State	$M_{R\delta t_u} = c$ (GAS3)		
	Emitter Time Error State	$M_{R\delta t_j} = -c$ (GAS4)		
	Link Propagation Error State	$M_{R\delta L} = 1$ (GAS5)		
	User Velocity Error State	$M_{RR\delta p} = r_j^T$ (GAS6)		
	Emitter Velocity Error State	Not required	$M_{RR\delta e_j} = -r_j^T$ (AS1)	
	User Clock Drift Error State	$M_{RR\delta t_u} = c$ (GAS7)		
	Emitter Clock Drift Error State	$M_{RR\delta t_j} = -c$ (GAS8)		
	Range Rate Error Due to User Position Error State	$M_{RR\delta p} = (\dot{p}+\dot{d})$ [D]** (G1)	$M_{RR\delta p} = (\dot{p}+\dot{d} - \dot{e}_j)^T[D]$ (AS2)	
	Range Rate Error Due to Emitter Position Error State	$M_{RR\delta e_j} = -M_{RR\delta p}$		

*Unless otherwise noted by an asterisk to indicate only one execution per emitter group.

$$**D = (I-r_j r_j^T)/|R_j|$$

TABLE XXIX. TMMM INPUT/OUTPUT SUMMARY

Inputs:

$$\left.\begin{array}{rl}
|R_j| & = \text{estimated scalar range to jth emitter (1x1)} \\
r_j & = \text{user/jth emitter unit LOS vector (3x1)} \\
(\dot{p}+\dot{d}) & = \text{antenna velocity w/r to fixed emitter (3x1)} \\
(\dot{p}+\dot{d}-\dot{e}_j) & = \text{antenna velocity vector w/r to moving emitter (3x1)}
\end{array}\right\} \text{TRRM}$$

Outputs: Measurement matrix elements defined according to the following convention:

$\overbrace{\text{Range-Diff. Measurement}}^{M^T}$	$\overbrace{\text{Range-Rate-Diff. Measurement}}^{M^T}$	Error States
$M_{R\delta p}$	$M_{RR\delta p}$	δp (= user position error)
$M_{R\delta e_j}$	$M_{RR\delta e_j}$	δe_j (= emitter position error)
$M_{R\delta t_u}$	0	δt_u (= user clock error)
$M_{R\delta t_j}$	0	δt_j (= emitter clock error)
$M_{R\delta L}$	0	δL (= link propagation error)
0	$M_{RR\delta p}$	δp
0	$M_{RR\delta e_j}$	δe_j
0	$M_{RR\delta t_u}$	δt_u
0	$M_{RR\delta t_j}$	δt_j

*In the KF Partitioned Structure Specification, the indices δp, δe_j, etc., are respectively denoted by the appropriate value of the KF error substate index s.

III.3.c.(8)

TRANSCEIVER DATA STATISTICS MODULE

(TDSM)

SPECIFICATION

The emitter data provided to the estimation algorithm within the Kalman filter requires an estimate of the measurement noise error covariance to properly weight the information. These noise statistics depend (a) upon the signal-to-noise ratio present within the tracking receiver bandwidth, and (b) upon tracking errors due to multipath conditions which present large deviations between the measured and extrapolated data. Other system noise error statistics which characterize the system error driving noises are also generated by this module.

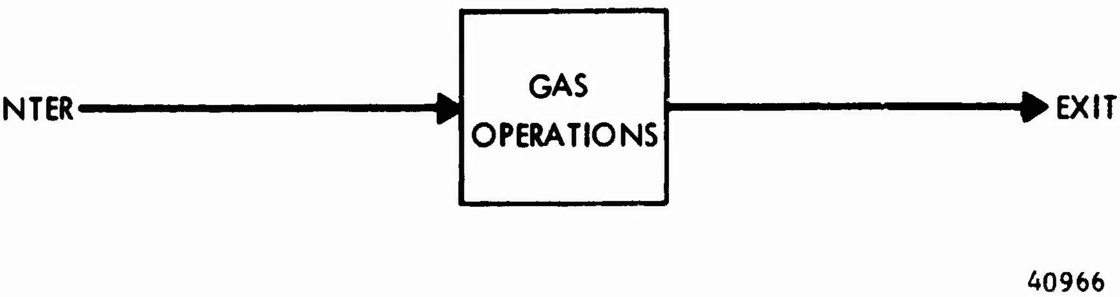


Figure 18. TDSM Operations Flow

TABLE XXX. TDSM OPERATIONS SUMMARY

	Operations \ Transceiver Configuration	Ground Emitter (G)	Airborne Emitter (A)	Satellite Emitter (S)	
Intra-KF-Cycle Execution (Once per Emitter Type j)*	Signal-to-Noise Ratio, α_j	$S/N = \alpha_j$ Receiver Input			(GAS1)
		$\alpha_j = K\alpha \frac{W_j}{\hat{R}_j^2}$ (Calculated)			(GAS2)
	Multipath Test Term, R_{Dj}	$\left(R_{mj} - \lvert R_j \rvert\right) = R_{Dj}$ $\left(\dot{R}_{mj} - \lvert \dot{R}_j \rvert\right) = \dot{R}_{Dj}$			(GAS3)
	Multipath Test Logic	$R_{Dj} > K_m \quad K_m^2 = N$ $R_{Dj} < K_m \quad K_m = 0$ $\dot{R}_{Dj} > \dot{K}_m \quad \dot{K}_m^2 = \dot{N}$ $\dot{R}_{Dj} < \dot{K}_m \quad \dot{K}_m = 0$			(GAS4)
	Emitter Range Variance, σ_{Rj}^2	$\sigma_{Rj}^2 = K_1 \alpha_j^{-1} + N$			(GAS5)
	Emitter Rate Variance, $\sigma_{\dot{R}j}^2$	$\sigma_{\dot{R}j}^2 = K_2 \alpha_j^{-1} + \dot{N}$			(GAS6)
	User Time Rate Variance, $\sigma_{\dot{t}u}^{2*}$	$\sigma_{\dot{t}u}^2 = K_3$			(GAS7)
	Emitter Time Rate Variance, $\sigma_{\dot{t}j}^{2*}$	$\sigma_{\dot{t}j}^2 = K_4$			(GAS8)
	Link Error Variance, σ_L^{2*}	$\sigma_L^2 = K_5$			(GAS9)
	Total Range Variance, C_{Rj}	$C_{Rj} = \sigma_{Gj}^2 + \sigma_L^2$			(GAS10)
	Total Range Rate Variance, C_{RRj}	$C_{RRj} = \sigma_{\dot{R}j}^2 + \sigma_{\dot{t}u}^2 + \sigma_{\dot{t}j}^2$			(GAS11)

*Unless otherwise noted by an asterisk to indicate execution once per emitter group.

TABLE XXXI. TDSM INPUT/OUTPUT SUMMARY

Inputs:

$|R_j|$ = estimated scalar range to jth emitter (1x1) } TRRM

$|\dot{R}_j|$ = estimated range rate to jth emitter (1x1) } TRRM

α_j = measured signal-to-noise ratio from receiver for jth emitter (optional input) (1x1)

W_j, K_5 = optional inputs from TEWM (see constants)

Outputs:

$\sigma^2_{R_j}$ = range measurement noise variance (1x1)

$\sigma^2_{\dot{R}_j}$ = range rate measurement noise variance (1x1)

$\sigma^2_{\dot{t}_u}$ = user time rate noise variance (1x1)

$\sigma^2_{\dot{t}_j}$ = emitter time rate noise variance (1x1)

σ^2_L = propagation link error noise variance (1x1)

C_{R_j} = total user/jth emitter range variance (1x1)

C_{RR_j} = total user/jth emitter range rate variance (1x1)

Constants Definitions:

W_j = transmitted emitter power for jth emitter obtained as a priori data for each emitter or optionally provided within the emitter data word

K_j = scaling constant for converting power ratio to signal-to-noise units

K_m = multipath range difference constant in range units, stored as a priori data

$\dot{K}_m$ = multipath rate difference constant in range rate units, stored as a priori data

N = multipath covariance term for range

$\dot{N}$ = multipath covariance term for range rate

K_1 = range covariance scaling of S/N or bandwidth

K_2 = range rate covariance scaling of S/N or bandwidth

K_3 = user clock offset rate error constant stored as a priori value

K_4 = emitter clock offset rate error constants stored as a priori value

K_5 = propagation link noise error constant stored as a priori data or provided as optional data word for emitter link

40984

III.4

KALMAN FILTER (K) MODULES

SPECIFICATIONS

Taken as a group, these modules together comprise all the operations necessary to (a) preselect and preprocess raw D/R measurement-differences into a form suitable for actual Kalman filtering (KMMM, KMRM, KMCM, KMOM); (b) accomplish Kalman filtering itself (KFIM); (c) accomplish estimate and covariance matrix prediction across the Kalman interval (between filtering times at the interval endpoints (KTUM, KTMM), and (d) compute and apply Kalman estimate-derived corrections to processor (Kalman and non-Kalman) navigation variables (KCOM).

All of these modules are partitioned into D, R, and D/R substate operations. Underlying these partitioned operations is the partitioned structure of the full processor error estimation model into D and R module-related partitions. This structure is therefore defined in detail in an initial, separate specification, distinct from the actual module specifications that follow, but necessary for interpreting them.

An understanding of the intermodular timing, sequencing and data flow is an equally necessary preliminary and reference for use of the module specifications themselves. Another initial, separate specification which defines these interrelationships, has therefore also been included.

III.4.a

KALMAN FILTER MODULES
PARTITIONED STRUCTURE

SPECIFICATIONS SUMMARY

This specification defines the D and R module-related, overall Kalman Filter structural partitioning by means of an ordered set of tables (Tables XXXII through XXXVII.

In particular, the initial Table XXXII defines the multilevel partitioning of the overall processor error state, from the two broadest-level partitions, which model to the processor error variables associated with the entire D module, and the entire R module groups respectively, to the finest level partitions, each of which models the error variables associated with only a submodular portion of just one of the D or R modules. Table XXXIII extends this partitioning beyond the error state vector to all the computational entities processed by the Kalman modules, at the broadest D and R group level.

Tables XXXIV through XXXVI then define the finest level, singly indexed partitioning (principally in terms of non-null partitions) of all these computational entities, for the D, R and D/R substates respectively, and Table XXXVII defines the doubly indexed (measurement-related) partitioning at this level.

It is important to note that these tables define substate reference frames, dimensions, indexing, and mnemonics which are used uniformly throughout all the Kalman modules.

TABLE XXXII. KF NAVIGATION ERROR SUBSTATE DEFINITIONS

		Substate Definition	Ref Frame	Dimension	Substate Index(s) IDR	PDR	ADR
Overall Navigation Error State	DR Nav Error Substates (D Substates)	User Position Error Substate	C	3	DI1	DP1	DA1
		User Velocity Error Substate	C	3	DI2	DP2	DA2**
		Platform/Computer Misalignment Substate	P	3	DI3	---	DA3
		Platf/Compr Mslgmt Rate Source Subst (Non-G-Sens)	P	*	DI4	---	DA4
		Platf/Compr Mslgmt Rate Source Subst (G-Sens)	P	*	DI5	---	DA5
		Specific Force Measurement Error Source Subst (Non-G-Sens)	P	*	DI6	---	---
		Specific Force Measurement Error Source Subst (G-Sens)	P	*	DI7	DP3	---
		Wind Error Source Substate	L	*	---	---	DA6
		TAS Error Source Substate	A	*	---	---	DA7
	Ref Nav Measurement Error Substates (R Substates)	User Ref Altitude Error Substate	----	*	RUA		
		User Clock Error Substate	----	*	RUC		
		⋮	⋮	⋮	⋮		
		nth Emtr Net Error Substate: nth Net Clock Error Substate	----	*	KEn: REnC		
		nth Net Ephemeris(Posn/Vel, etc.) E. S.	C	*	REnE		
		⋮	⋮	⋮	⋮		
		nth Net kth Signal jth Emtr Error Substate	----	*	REnjk		
		⋮	⋮	⋮	⋮		
		⋮	⋮	⋮	⋮		

* Dimension depends on equipment types and KF model depth required.

40985

** Pseudosubstate whose use facilitates DR mode switching and provides direct velocity error statistics in ADR mode.

TABLE XXXIII. KF D/R SUBSTATE LEVEL STRUCTURE

Error State

$$x = \begin{bmatrix} x_D \\ x_R \end{bmatrix}$$

Error State Control

(Impulsive)

$$u = \begin{bmatrix} u_D \\ u_R \end{bmatrix}$$

(Metered)

$$\dot{u} = \begin{bmatrix} \dot{u}_D \\ 0 \end{bmatrix}$$

Error State Dynamic Coefficient Matrix

$$A = \begin{bmatrix} A_D & 0 \\ 0 & A_R \end{bmatrix}$$

Error State Transition Matrix

$$\phi = \begin{bmatrix} \phi_D & 0 \\ 0 & \phi_R \end{bmatrix}$$

Error State Metered Control Distribution Matrix

$$G = \begin{bmatrix} G_D & 0 \\ 0 & 0 \end{bmatrix}$$

Error State Noise PSD Matrix

$$K^{**} = \begin{bmatrix} K_D^{**} & 0 \\ 0 & K_R^{**} \end{bmatrix}$$

Error State Noise Increment Matrix

$$R^{*} = \begin{bmatrix} R_D^{*} & 0 \\ 0 & R_R^{*} \end{bmatrix}$$

Error State Error Covariance Matrix

$$P^{*} = \begin{bmatrix} P_D^{*} & P_{D/R} \\ P_{D/R}^{T} & P_R^{*} \end{bmatrix}$$

Synchronous Measurement Diff.

$$\Delta Y = Y_D - Y_R$$

Measurement/State Matrix

$$M = \begin{bmatrix} M_D & M_R \end{bmatrix}$$

Measurement Diff. Noise

$$C = C_D + C_R$$

Kalman Optimal Gain Matrix

$$b_K = \begin{bmatrix} b_{KD} \\ b_{KR} \end{bmatrix}$$

Least Squares Gain Matrix

$$b_L = \begin{bmatrix} b_{LD} \\ b_{LR} \end{bmatrix}$$

Overall Gain Matrix

$$b = \begin{bmatrix} b_D \\ b_R \end{bmatrix}$$

Time-Averaged Measurement-Diff.

$$\overline{\Delta Y} = \overline{Y}_D - \overline{Y}_R$$

Time-Averaged Measurement/State Matrix

$$\overline{M} = \begin{bmatrix} \overline{M}_D & \overline{M}_R \end{bmatrix}$$

Time-Averaged Measurement Control Compensation Matrix

$$\overline{N} = \begin{bmatrix} \overline{N}_D & 0 \end{bmatrix}$$

Time-Averaged Measurement State Noise Compensation Matrices

$$\overline{Z} = \begin{bmatrix} \overline{Z}_D & \overline{Z}_R \end{bmatrix}$$

$$\overline{W} = \overline{W}_D + \overline{W}_R$$

Dimension Summary

nxl: $x, u, \dot{u}, b_K, b_L, b$

nxn: A, ϕ, G, K, R, P

lxn: $M, \overline{M}, \overline{N}, \overline{Z}$

lxl: $Y, \Delta Y, C, \overline{\Delta Y}, \overline{Y}, \overline{C}, \overline{W}$

Time-Averaged Measurement-Diff. Noise

$$\overline{C} = \overline{C}_D + \overline{C}_R$$

*Symmetric Matrices

**Diagonal Matrices

40986

TABLE XXXIV. D SUBSTATE VECTOR/MATRIX STRUCTURE

Overall D State Vector or Matrix	D Substate Vector or Matrix	Dimensions (n_{Ds} = D Substate Dim)	Special Structural Properties	Structure by DR Nav Mode (Non-Null D Partitions) IDR(DI)	PDR(DP)	ADR(DA)
x_D	x_{Ds}	$n_{Ds} \times 1$	----	All	All	All
u_D	u_{Ds}	"	----	"	"	"
$\dot{u}_D$	$\dot{u}_{Ds}$	"	----	3	---	---
b_{KD}	b_{KDs}	"	----	All	All	All
b_{LD}	b_{LDs}	"	----	"	"	"
b_D	b_{Ds}	"	----	"	"	"
A_D	$A_{Dss'}$	$n_{Ds} \times n_{Ds'}$	----	12,21,22,23,26,27,33,34, 35,44,55,66,77	12,23,33	13,16,17,31,33 -37,44, 55,66,77
K_D	K_{Ds}	$n_{Ds} \times n_{Ds}$	K_D, K_{Ds} Diag.	4-7	3	4-7
Φ_D	$\Phi_{Dss'}$	$n_{Ds} \times n_{Ds'}$	----	11-17,21-27,33-35,44,55, 66,77	11-13,22-23,33	11,13-17,31,33-37,44, 55,66,77
G_D	$G_{Dss'}$	"	----	13,23,33	---	---
R_D	$R_{Dss'}$	"	R_D Symm.	11-17,22-27,33-35,44, 55,66,77 and Symm.	All	11,13-17,33-37,44,55,66, 77 and Symm.
P_D	$P_{Dss'}$	"	P_D Symm.	All	All	All

40987

$A_{DI12} = I$, $A_{DI21} = -\omega_s^2\left(I - 3\frac{g}{|g|}\frac{g}{|g|}^T\right)$, $A_{DI22} = -2\omega_{E/I}^x$, $A_{DI23} = T_{C/P} f_P^x$, $A_{DI26} = T_{C/P} Q_{DI26}$

$A_{DI27} = T_{C/P} Q_{DI27}$, $A_{DI33} = -\left(\omega_{P/I}\right)_P^x$, $A_{DI34} = Q_{DI34}$, $A_{DI35} = Q_{DI35}$, $A_{DI44} = Q_{DI44}$, $A_{DI55} = Q_{DI55}$

$A_{DI66} = Q_{DI66}$, $A_{DI77} = Q_{DI77}$, $A_{DP12} = I$, $A_{DP23} = T_{L/C} I$, $A_{DP33} = -K_{\beta L}$

(A_{DAs} to be determined)

(Unspecified Qs and Ks depend on DR nav equipment types and on depth of KF error model required)

TABLE XXXV. R SUBSTATE VECTOR/MATRIX STRUCTURE

Overall R State Vector or Matrix	R Substate Vector or Matrix	Dimensions (n_{Rs} = Substate Dimension)	Special Structural Properties	Structure (Non-Null R Partitions)
x_R	x_{Rs}	$n_{Rs} \times 1$	----	All
u_R	u_{Rs}	"	----	"
b_{KR}	b_{KRs}	"	----	"
b_{LR}	b_{LRs}	"	----	"
b_R	b_{RS}	"	----	"
K_R	K_{Rs}	$n_{Rs} \times n_{Rs}$	K_R, K_{Rs} Diag	All Pseudodiag.
A_R	A_{Rs}	"	A_R Pseudodiag.*	"
ϕ_R	ϕ_{Rs}	"	ϕ_R Pseudodiag.	"
R_R	R_{Rs}	"	R_R Pseudodiag., R_{Rs} Symm	"
P_R	$P_{Rss'}$	$n_{Rs} \times n_{Rs'}$	P_R Symm.	All

*All off-diagonal partitions of a pseudodiagonal matrix are null.

(As and Ks depend on Ref Nav measurement equipment types and on KF model depth required).

TABLE XXXVI. D/R SUBSTATE VECTOR/MATRIX STRUCTURE

Overall D/R Coupling Matrix	D/R Substate Matrix	Dimensions	Special Structural Properties	Structure (Non-Null D/R Partitions)
$P_{D/R}$	$P_{D/Rss'}$	$n_{Ds} \times n_{Rs'}$	$P_{D/R} = P_{R/D}^T$	All
$P_{R/D}$	$P_{R/Dss'}$	$n_{Rs} \times n_{Ds'}$	$P_{R/D} = P_{D/R}^T$	All

TABLE XXXVII D/R MEASUREMENT-DIFFERENTIAL VECTOR/MATRIX STRUCTURE

	Overall Vector or Matrix	Sub-State	Dimensions	Structure (Non-Null Partitions) Pos'n Compon't(m=1)	Altitude(m=2)	LOS Range* (m=3)	LOS R Rate* (m=4)	EM Range* (m=5)
D	Y_{Dm}	----	1 x 1	---	---	---	---	---
	$\dot{Y}_{Dm}$	----	"	---	---	---	---	---
	C_{Dm}	----	"	---	---	---	---	---
	$\dot{C}_{Dm}$	----	"	---	---	---	---	---
	M_{Dm}	M_{Dms}	$1 \times n_{Ds}$	DI1/DP1/DA1			DI1,2/DP1,2/DA1, 2,6,7	DI1/DP1/DA1
	$\tilde{M}_{Dm}$	$\tilde{M}_{Dms}$	"	DI1-DI7/DP1-DP3/DA1,DA3-DA7				
	$\tilde{Z}_{Dm}$	$\tilde{Z}_{Dms}$	"					
	$\bar{R}_{Dm}$	$\bar{R}_{Dms}$	"	DI4/ - / -				
	$\bar{W}_{Dm}$	$\bar{W}_{Dms}$	1 x 1	---	---	---	---	---
R	Y_{Rm}	---	1 x 1	---	---	---	---	---
	$\dot{Y}_{Rm}$	---	"	---	---	---	---	---
	C_{Rm}	---	"	---	---	---	---	---
	$\dot{C}_{Rm}$	---	"	---	---	---	---	---
	M_{Rm}	M_{Rms}	$1 \times n_{RS}$	None	RuA	RuC;REnC,REnE,REnjk		
	$\tilde{M}_{Rm}$	$\tilde{M}_{Rms}$	"					
	$\tilde{Z}_{Rm}$	$\tilde{Z}_{Rms}$	"					
	$\bar{W}_{Rm}$	$\bar{W}_{Rms}$	1 x 1	---	---	---	---	---
D/R	ΔY_m	---	1 x 1	---	---	---	---	---
	$\Delta\dot{Y}_m$	---	"	---	---	---	---	---

*kth signal of jth emitter of nth net

40988

$Y_{Dm} = p, |h|, |R_{nj}|, \dot{R}^T_{nj} r_{nj}, S_{nj}$

$Y_{Rm} = p_R, |h_R|, R_{Mnj}, \dot{R}_{Mnj}, S_{Mnj}$

(Undefined Ms and Cs depend on nav equipment types and KF model depth required.)

$r_{nj} = R_{nj}/|R_{nj}| \quad R_{nj} = p - c_{nj}$

$m = 1 \quad M_{DI1} = M_{DP1} = M_{DA1} = c_i^T$

$m = 2 \quad M_{DI1} = M_{DP1} = M_{DA1} = -g^T/|g|;\ M_{RUA}$

$m = 3 \quad M_{DI1} = M_{DP1} = M_{DA1} = r^T_{nj};\ M_{RUc}, M_{REnC} M_{REnE}, M_{REnjk}$

$m = 5 \quad M_{DI1} = M_{DP1} = M_{DA1} = t_{nj};\ M_{RUC}, M_{REnC}, M_{REnE}, M_{REnjk}$

$m = 4 \quad M_{DI2} = M_{DP2} = r^T_{nj}, M_{DA2}, M_{DA5}, M_{DA6}$

$M_{DI1} = M_{DP1} = -\left(\dot{R}^T_{nj}/|R_{nj}|\right)\left(1 - r_{nj} r^T_{nj}\right), M_{DA1}$

$M_{RUC}, M_{REnC}, M_{REnE}, M_{REnjk}$

III.4.b

KALMAN FILTER MODULES

TIMING AND SEQUENCING ORGANIZATION

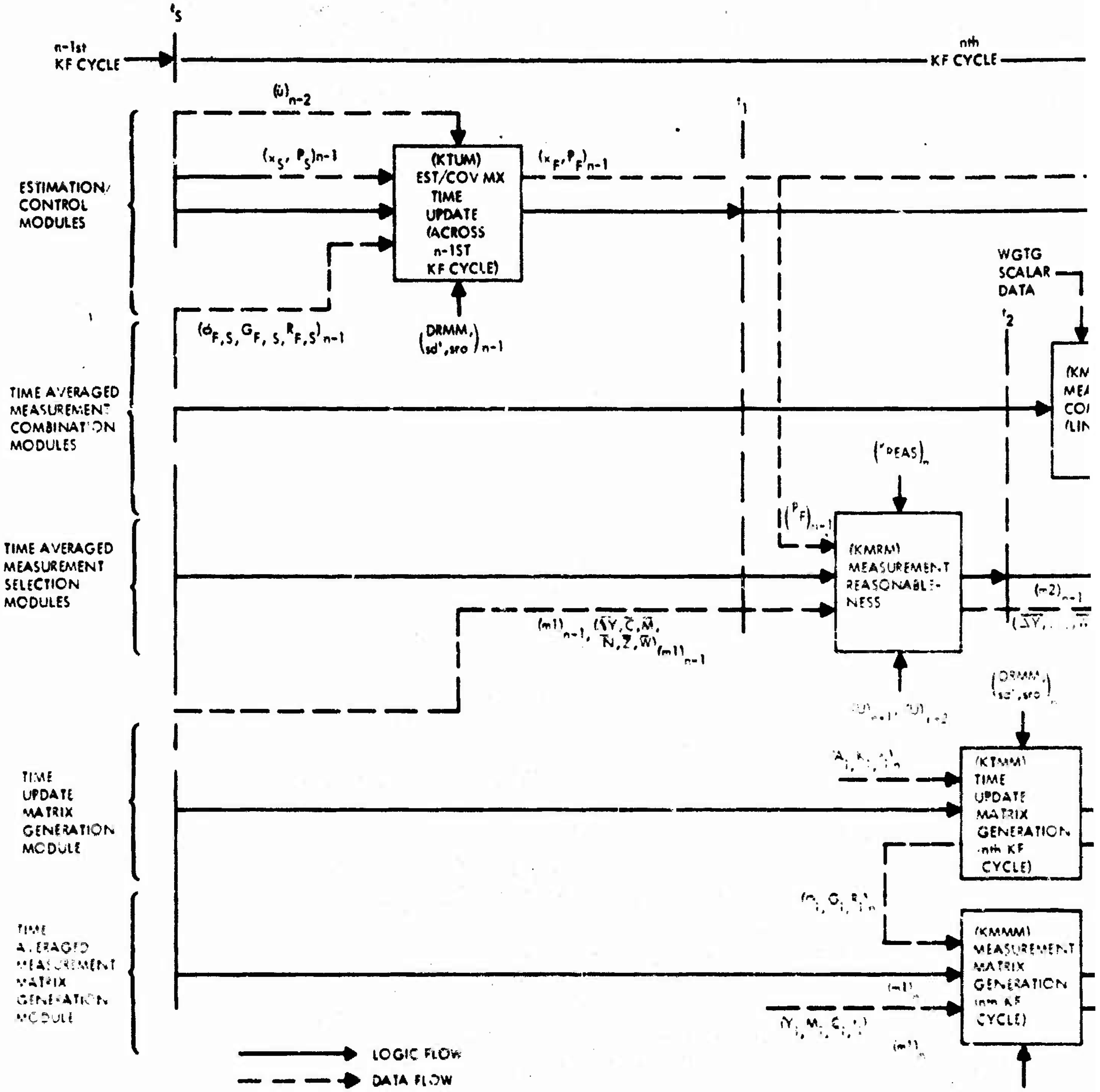
This specification defines the relative intermodular timing, sequencing, and data flow for all Kalman filter modules, within a single, standard (i.e., all-operations), current (nth), Kalman filter execution cycle.

The summary diagram (Figure 19) divides overall single-cycle filter operations into two more or less parallel processing areas. The first of these sequentially involves (a) KTUM prediction of the processor error estimate and covariance matrix across the prior cycle, using prediction matrices generated by the KTMM in the prior cycle; (b) KMRM, KMCM, and KMOM preprocessing of raw measurements collected, time-smoothed and synchronized during the prior cycle by the KMMM (and/or R modules) for current-cycle use by the KFIM; (c) actual Kalman filtering by the KFIM using these preprocessed measurements, and (d) the KCOM generation, and end-of-current-cycle execution, of processor module (D, R, and K) control corrections, based on the updated estimate obtained from KFIM filtering.

The second of these parallel processing areas involves (a) KTMM generation of current-cycle prediction matrices for next-cycle KTUM use, and (b) KMMM generation of current-cycle, endpoint-synchronized and time-smoothed measurements and measurement matrices for use by the measurement preprocessing modules (KMRM, KMCM, and KMOM) in the next cycle.

In general, the overall diagram for clarity depicts only full-state operations; however, each of the module specifications themselves in fact defines the partitioned equivalents of these full-state operations, in terms of indexed operations on the D and R substate sets (sd, sd', sri), as defined at the bottom of the diagram.

In addition, the measurement sets remaining after successive execution of each of the measurement preprocessing module operations are also symbolized and defined at the bottom of the diagram. These measurement set symbols are used in defining doubly indexed substate processing operations by measurement type in the measurement processing (i.e., KFIM, KMRM, KMCM, KMOM, and KMMM) modules.



Note

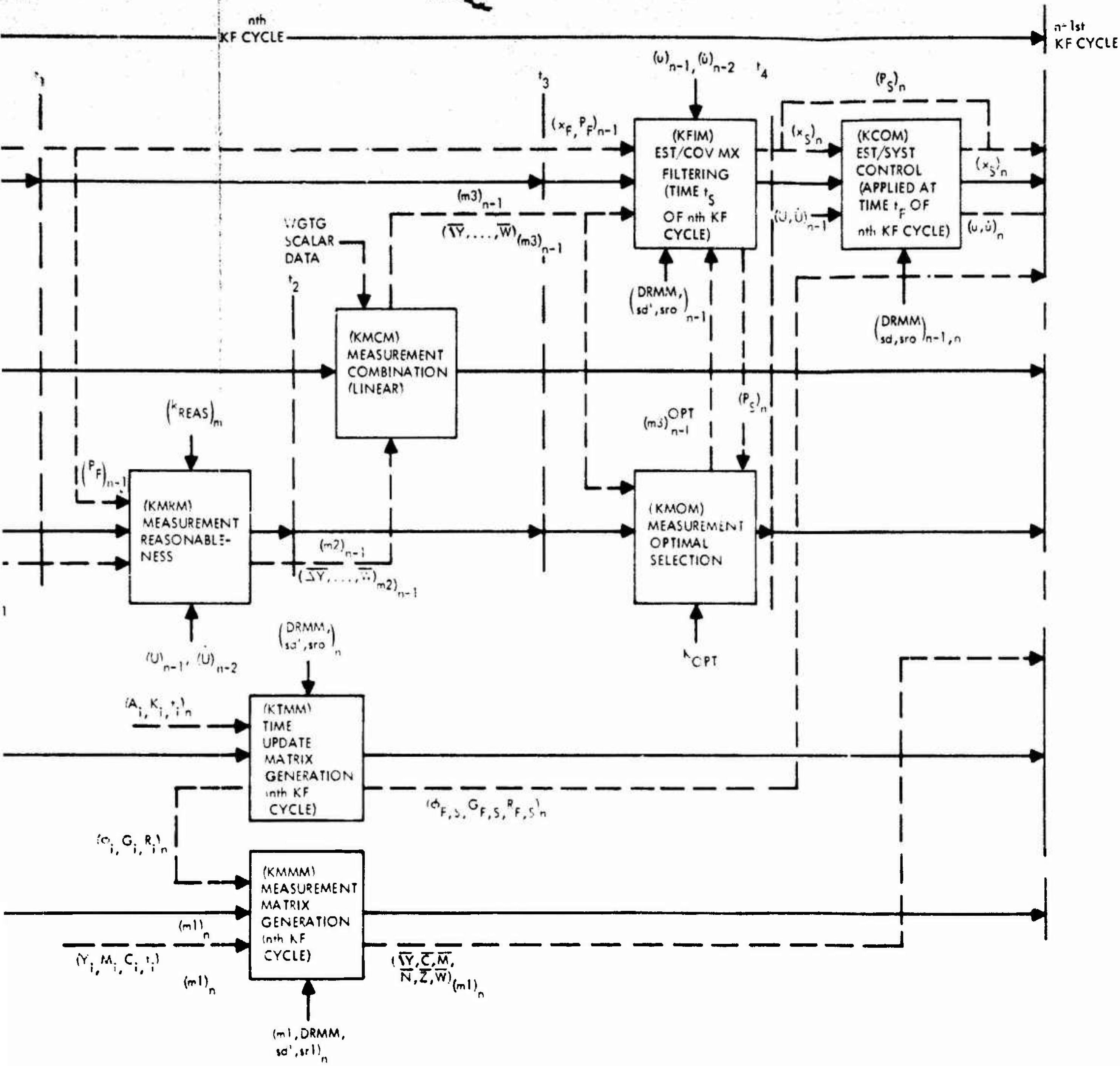
If additional programming flexibility in a particular application is required, completion of current-cycle KMMM operations could in fact be delayed until time t_1 of the next cycle. However, the effect of such a timing change on the overall KF switching specifications (KSWM) would need careful review and appropriate revision.

sd = OVERALL SET OF KFD SUBSTATES* (DR MODE DEPENDENT)

sd' = sd, BUT WITH SUBSTATES DI1, DI2; DP1, DP2; DA1, DA3 RESPECTIVELY COMBINED INTO THE SINGLE SUBSTATES DI1 + DI2; DP1 + DP2; DA1 + DA3

*EXCLUDING ADR PSEUDOSUBSTATE DA2

Figure 19. Kalman Filter Modules, Overall Dynamic Operations Timing/Sequencing/Data Requirements (Standard KF Cycle)



...LL SET OF KFD SUBSTATES* (DR MODE DEPENDENT)
...T WITH SUBSTATES DI1, DI2;
...DP2; DA1, DA3 RESPECTIVELY
...NED INTO THE SINGLE SUBSTATES
...DI2, DP1 + DP2; DA1 + DA3

...O ADD PSEUDOSUBSTATE DA2

mo OVERALL SET OF CURRENTLY AVAILABLE REF NAV MEASUREMENT OUTPUT DATA TYPES

m1 SUBSET OF mo PRESELECTED BY OPERATOR IN LAST CYCLE FOR KMMM PROCESSING IN CURRENT CYCLE

m2 SURVIVING SUBSET OF m1 AFTER REASONABLENESS TESTING

m3 SET CONSISTING OF LINEAR (OR NONLINEAR) COMBINATIONS OF m2

sri SET OF KF R SUBSTATES CORRESPONDING TO REF NAV MEASUREMENT SET mi

...Operations
...ndard KF Cycle)

4035b

III.4.c

KALMAN FILTER ESTIMATE/COVARIANCE MATRIX
TIME UPDATE MODULE

(KTUM)

SPECIFICATION

This module time-updates the KF error state estimate vector and its associated covariance matrix, across the prior cycle to the beginning of the current cycle (time t_S), using the time update matrices generated by the KTMM in the prior cycle.

TABLE XXXVIII. KTUM OPERATIONS SUMMARY

Operation	Substate Class	D Substates	R Substates	D/R Substates
Execute Once per KF Cycle	Substate Index Set-up	$s,s',i,j = sd'$ (sd' DR Mode Dependent)*	$s,s' = sr0$	$s=sd',\ s'=sr0$
	Est/Cov. Matrix Substates Time Update (Across Prior Cycle)	$x_{Ds} = \sum_j \left(\Phi_{Dsj} x_{Dj} + G_{Dsj} \dot{u}_{Dj}\right)$ $P_{Dss'} = \sum_i \sum_j \left(\Phi_{Dsi} P_{Dij} \Phi^T_{Djs'}\right) + R_{Dss'}$ (All s,s',i,j)*	$x_{Rs} = \Phi_{Rs} x_{Rs}$ $P_{Rss'} = \Phi_{Rs} P_{Rss'} \Phi^T_{Rs'} + \delta_{ss'} R_{Rs}$ (δ=Kronecker delta) (All s,s')	$P_{D/Rss'} = \left(\sum_i \Phi_{Dsi} P_{D/Ris'}\right) \Phi^T_{Rs'}$ (All s,s')*

*For ADR, additional DA2 pseudostate updating is required (executed just following the above time updating) as follows:

$$x_{D2} = \sum_j H_{vAj} x_{DAj}$$

$$P_{D2s'} = \sum_j H_{vAj} P_{Djs'} \qquad (j = \text{All ADR Substates})$$

$$P_{D/R2s'} = \sum_j H_{vAj} P_{D/Rjs'}$$

where H_{vAj} are submatrices of $H_{vA} = \left[0\ 0\ H_{vA3}\ 0\ 0\ H_{vA6}\ H_{vA7}\right]$

and $H_{vAj} = A_{DA1j}$ (at time t_F of n-1st KF cycle for nth KF cycle operations)

TABLE XXXIX. KTUM INPUT/OUTPUT SUMMARY

Inputs/Outputs \ Substate Class	D Substates	R Substates	D/R Substates
Inputs	$\underline{s,s' = (sd')_{n-1}}$ $(\phi_{Dss'}, G_{Dss'}, R_{Dss'})(F,S)n-1$; $(x_{Ds}, P_{Dss'})_{S,n-1}$; $(\dot{u}_{Ds})_{n-2}$ $\underline{s=(DA2)_{n-1}, s'=(sd',DA2)^{*}_{n-1}}$ $(H_{vAs'}, P_{Dss'})_{F,n-1}$	$\underline{s,s' = (sr0)_{n-1}}$ $(\phi_{Rs}, R_{Rs})(F,S)n-1$; $(x_{Rs}, P_{Rss'})_{S,n-1}$	$\underline{s = (sd',sr0)_{n-1}}$ $(P_{D/Rss'})_{S,n-1}$ $\underline{s=(DA2)_{n-1}, s'=(sr0)_{n-1}}$ $(P_{D/Rss'})_{F,n-1}$
Outputs	$\underline{s,s' = (sd')_{n-1}}$ $(x_{Ds}, P_{Dss'})_{F,n-1}$ $\underline{s=(DA2)_{n-1}, s'=(sd',DA2)^{*}_{n-1}}$ $(x_{Ds}, P_{Dss'})_{F,n-1}$	$\underline{s,s' = (sr0)_{n-1}}$ $(x_{RS}P_{Rss'})_{F,n-1}$	$\underline{s = (sd',sr0)_{n-1}}$ $(P_{D/Rss'})_{F,n-1}$ $\underline{s=(DA2)_{n-1}, s'=(sr0)_{n-1}}$ $(P_{D/Rss'})_{F,n-1}$

*Additional Input/Output requirements for ADR only.

III.4.d

KALMAN FILTER MEASUREMENT PREPROCESSING MODULES

(1) MEASUREMENT REASONABLENESS TESTING (KMRM)

(2) MEASUREMENT OPTIMAL SELECTION (KMOM)

(3) MEASUREMENT COMBINATION (KMCM)

SPECIFICATIONS

These modules respectively reasonableness-edit (KMRM), combine (KMCM), and order (KMOM) the time-smoothed, endpoint-synchronized measurements from last-cycle KMMM (and R module) operations, in preparation for their use by the KFIM module in the current cycle.

In particular, it should be noted that the KMCM (measurement combination module) specification given here (and the related KF measurement type definitions of subsection III.4.a), explicitly accommodate only linear combinations of measurements of different types, and not the special start-up, nonlinear LOS measurement combination technique presented and discussed in detail elsewhere in this document (see subsection III.g.3 and Appendix IX).

Although it appears that this specification could be quite simply modified to include this capability as well, this has not been done since it is felt that this promising technique merits a further, well-rounded, overall investigation of its own -- one which would in particular include a deeper examination of its important statistical, geometrical, time-sequencing, and equipment-requirement aspects.

Finally, it should be noted that the KMMM specification here (in conjunction with the measurement type definitions of subsection III.4.a) implies that the operations of raw measurement D/R differencing, KF endpoint synchronization, and time smoothing must be done collectively within the context of Kalman operations -- i.e., under Kalman timing control. However, these operations could alternatively -- and perhaps preferably -- be placed in the context of, and within the timing control of only the pertinent R modules instead. As a third alternative in certain circumstances (e.g., an R measurement which is available at a very high rate) they could perhaps best be placed under D module timing control. Which of these options is preferable in programming for a specific application will depend in general on factors peculiar to that application. To cover the two most likely possibilities, these operations have therefore in fact been specified in two places -- i.e., in both the K and the R module groups -- in this document.

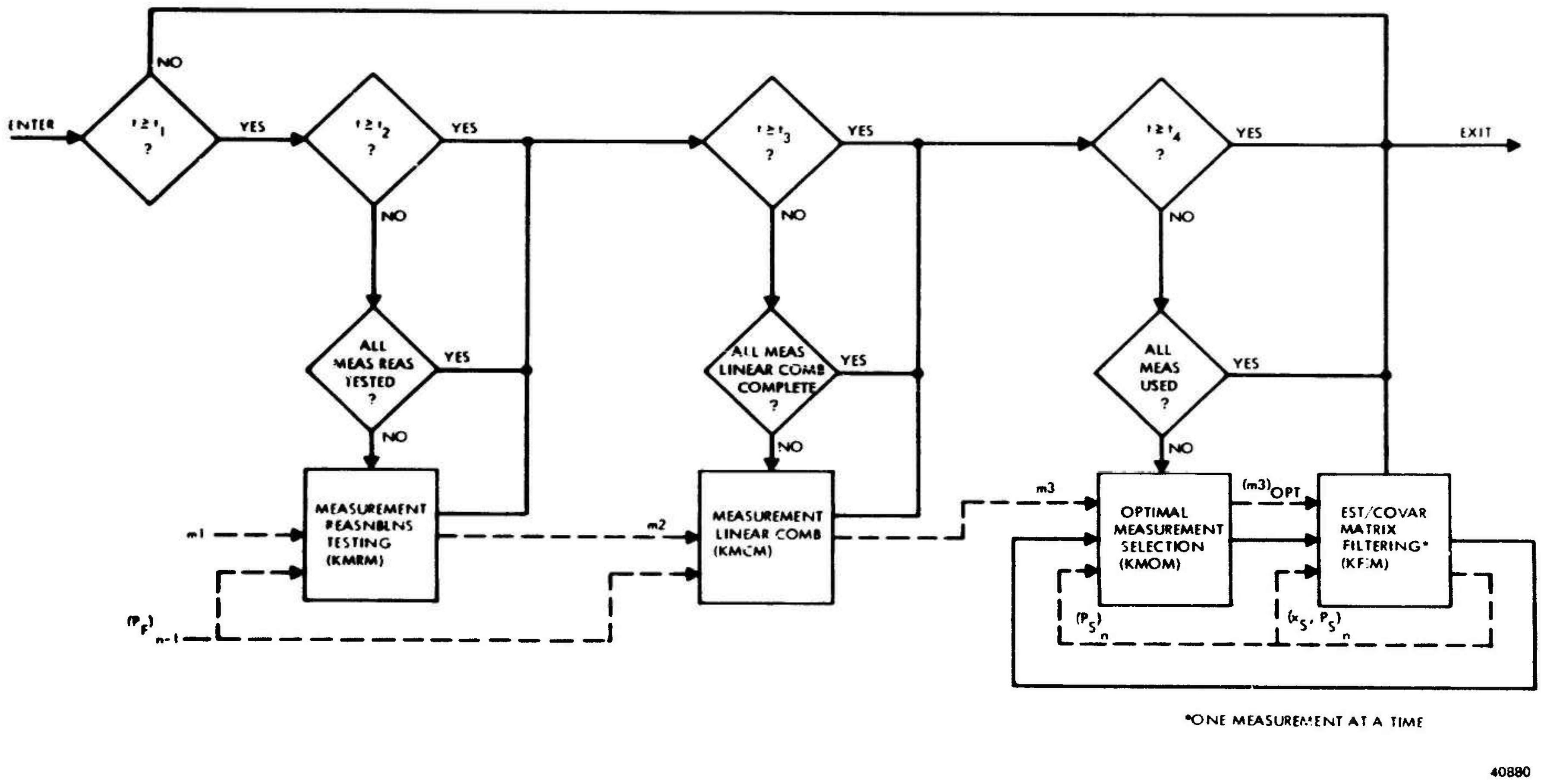


Figure 20. KMRM, KMCM, KMOM and KFIM Submodule Logic/Data Flow

TABLE XL. KMRM OPERATIONS SUMMARY

Operation (rows) \ Substate Class (columns)		D Substates	R Substates	D/R Substates
Input Measurement Type Index Setup		$m = m1$		
Execution Once per KF Cycle per Averaged Measurement Type m1	Substate Index Setup	$s, s' = sd'$ (sd' DR Mode Dependent)	$s, s' = sr1$	--------
	Unbiased Measurement Computation	*	*	*
	Measurement Residual Variance Computation	*		
	Averaged Measurement Reasonableness Test	$\left(\overline{\Delta Y'_m}\right)^2 \le \left(k_{REAs}\right)^2_m \bar{Q}_m$ (Measurement m is reasonable)		
		" > " (Measurement m is <u>not</u> reasonable)		

*Same Equations as for Estimate/Covariance Matrix Filtering Submodule.

TABLE XLI. KMOM OPERATION SUMMARY

Equation (rows) \ Substate Class (columns)		D Substates	R Substates	D/R Substates
Execution Once per KF Cycle per Averaged Combined Measurement Type m3	Input Measurement Type Index Setup	$m = m3$		
	Trace Computations	TR_m = Trace of $K_{OPT} \Delta P_{Dm} K_{OPT}^T$ (All m)		
	Optimum Measurement Identification	Find m such that TR_m = MAX (All m)		

If only the 1' (position and velocity error) partition of K_{OP1} are non-null (which will often be the case), then the trace computation algorithm above reduces to:

$$TR_m = \text{TRACE OF } (K_{OPT1'} \Delta P_{D1'1'm} K_{OPT1'})$$

TABLE XLII. KMCM OPERATIONS SUMMARY

<table>
<tr><th colspan="2">Substate Class / Operation</th><th>D Substates</th><th>R Substates</th></tr>
<tr><td rowspan="2">Execution Once per KF Cycle</td><td>Measurement Type Index Setup</td><td colspan="2">$m = m3,\ m' = m2$</td></tr>
<tr><td>Substate Index Setup</td><td>$s = sd'$
(DR Mode Dependent)</td><td>$s = sr2$</td></tr>
<tr><td rowspan="4">Execution Once per KF Cycle for each Combined Measurement Type m3</td><td>Scalar Weight ($w_{mm'}$) Computations</td><td colspan="2">$w_{mm'}$ = Depends on Candidate Measurement Combination Technique</td></tr>
<tr><td>$\overline{\Delta Y}'_m, \overline{C}_m$ Computations</td><td>$\overline{\Delta Y}'_m = \sum_{m'} w_{mm'} \overline{\Delta Y}'_{m'}$</td><td>$\overline{C}_m$ = Function* of $\overline{C}_{m'}$'s</td></tr>
<tr><td>$\overline{M}_{ms}, \overline{N}_{ms}$ Computations</td><td>$\overline{M}_{Dms} = \sum_{m'} w_{mm'} \overline{M}_{Dm'}$'s
$\overline{N}_{Dms} = \sum_{m'} w_{mm'} \overline{N}_{Dm'}$'s</td><td>$\overline{M}_{Rms} = \sum_{m'} w_{mm'} \overline{M}_{Rm'}$'s</td></tr>
<tr><td>$\overline{Z}_{ms}, \overline{W}_{ms}$</td><td>*</td><td>*</td></tr>
</table>

*Depends on Candidate Measurement Combination Algorithm

TABLE XLIII. KMRM INPUT/OUTPUT SUMMARY

Inputs/Outputs \ Substate Class	D Substates	R Substates	D/R Substates
Inputs	$\underline{m=(ml)_{n-1}; s, s'=(s'_d)_{n-1}}$ $(\overline{Y}_{Dm}, \overline{C}_{Dm}, \overline{M}_{Dms}, \overline{N}_{Dms}, \overline{Z}_{Dms}, \overline{W}_{Dms})_{n-1}, (\hat{u}_{Ds})_{n-2}, (P_{Dss'})_{F,n-1}$	$\underline{m=(ml)_{n-1}; s=(srl)_{n-1}}$ $(\overline{Y}_{Rm}, \overline{C}_{Rm}, \overline{M}_{Rms}, \overline{Z}_{Rms}, \overline{W}_{Rms})_{n-1}, (P_{Rss'})_{F,n-1}$	$\underline{m=(ml)_{n-1}}$ $(k_{REAs})_m$
Outputs	$\underline{m=(ml)_{n-1}}$ $(\overline{Q}^{*}_{Dm}, \overline{Y}'_{Dm})_{n-1}$	$\underline{m=(ml)_{n-1}}$ $(\overline{Q}^{*}_{Rm}, \overline{Y}'_{Rm})_{n-1}$	$\underline{m=(ml)_{n-1}}$ $(\overline{Q}^{*}_{D/Rm}, \overline{Q}^{*}_{m}, \overline{\Delta Y}_{m})_{n-1}, (m2)_{n-1}$

*Q's for Reasonableness Testing only; all are based on <u>same</u> $(P_F)_{n-1}$

TABLE XLIV. KMOM INPUT/OUTPUT SUMMARY

Substate Class / Inputs/Outputs	D Substates Only
Inputs	K_{OPT}, ΔP_{Dm}
Outputs	TR_m, $(TR_m)_{max}$

TABLE XLV. KMCM INPUT/OUTPUT SUMMARY

Inputs/Outputs \ Substate Class	D Substates	R Substates	D/R Substates
Inputs	$\underline{m=(m2)_{n-1}, s=(sd')_{n-1}}$ $(\bar{M}_{Dms}, \bar{N}_{Dms}, \bar{Z}_{Dms}, \bar{W}_{Dms})_{n-1}$	$\underline{m=(m2)_{n-1}, s=(sr2)_{n-1}}$ $(\bar{M}_{Rms}, \bar{Z}_{Rms}, \bar{W}_{Rms})_{n-1}$	$\underline{m=(m3)_{n-1}, m'=(m2)_{n-1}}$ $(\bar{w}_{mm'})_{n-1}$ Definition Inputs, $(\overline{\Delta Y'}_{m'})_{n-1}$
Outputs	$\underline{m=(m3)_{n-1}, s=(sd')_{n-1}}$ $(\bar{M}_{Dms}, \bar{N}_{Dms}, \bar{Z}_{Dms}, \bar{W}_{Dms})_{n-1}$	$\underline{m=(m2)_{n-1}, s=(sr2)_{n-1}}$ $(\bar{M}_{Rms}, \bar{Z}_{Rms}, \bar{W}_{Rms})_{n-1}$	$\underline{m=(m3)_{n-1}, m'=(m2)_{n-1}}$ $(w_{mm'}, \overline{\Delta Y}'_m, \bar{C}_m)_{n-1}$

III.4.e

KALMAN FILTER ESTIMATE/COVARIANCE MATRIX
FILTERING UPDATE MODULE

(KFIM)

SPECIFICATION

This module summarizes the operations which comprise actual Kalman filtering of the preprocessed and preselected measurements; i.e., the generation of a new, overall processor error estimate (and its attendant covariance matrix) using these measurements and the a priori estimate generated by the KTUM in the current cycle. Both estimates -- the KTUM current cycle estimate and the one generated by this module in the current cycle -- are estimates of processor error state at time t_s of the current cycle, but the latter, measurement-improved estimate supersedes and replaces the former.

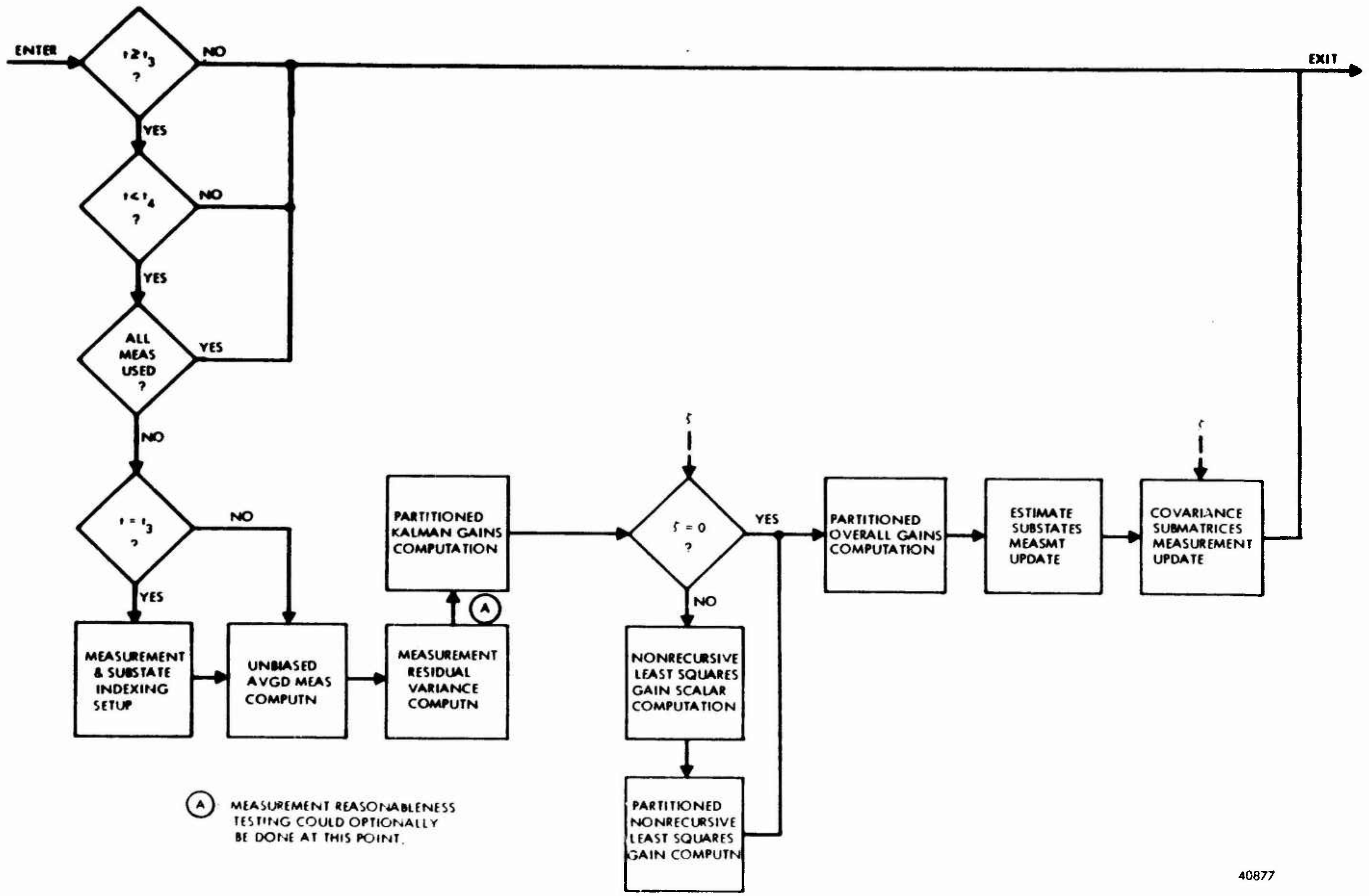


Figure 21. Estimate/Covariance Matrix Substates Measurement Update Logic Flow

TABLE XLVI. KFIM OPERATIONS SUMMARY

Operation	Substate Class	D Substates	R Substates	D/Ṙ Substates
Execution Once per KF Cycle	Input Measurement Type Index Setup	m = m1 or m2 or m3*		
Execution Once per KF Cycle	Substate Index Setup	s,s' = sd' (sd' DR Mode dependent)	s,s' = sr0	s = sd', s' = sr0
Execution once per KF Cycle per Averaged Measurement Type m1 or m2 or m3	Measurement Residual Variance Computation.	$\bar{Q}_{Dm}=\sum_{s}\sum_{s'}\left(\bar{M}_{Dms}P_{Dss'}\bar{M}_{Dms'}^{T}\right)+\bar{C}_{Dm}+\sum_{s}\left(\bar{W}_{Dms}-2\bar{M}_{Dms}\bar{Z}_{Dms}^{T}\right)$	$\bar{Q}_{Rm}=\sum_{s}\sum_{s'}\left(\bar{M}_{Rms}P_{Rss'}\bar{M}_{Rms'}^{T}\right)+\bar{C}_{Rm}+\sum_{s}\left(\bar{W}_{Rms}-2\bar{M}_{Rms}\bar{Z}_{Rms}^{T}\right)$	$\bar{Q}_{D/Rm}=\sum_{s}\sum_{s'}\left(\bar{M}_{Dms}P_{D/Rss'}\bar{M}_{Rms'}^{T}\right)$
		$\bar{Q}_m=\bar{Q}_{Dm}+\bar{Q}_{Rm}+2\bar{Q}_{D/Rm}$		
	Nonrecursive Least Squares Weighting Scaler Computation	$\bar{\mu}_{Dm}=\sum_{s}\left(\bar{M}_{Dms}\bar{M}_{Dms}^{T}\right)$	$\bar{\mu}_{RM}=\sum_{s}\left(\bar{M}_{Rms}\bar{M}_{Rms}^{T}\right)$	----------------
		$\bar{\mu}_m=\bar{\mu}_{Dm}+\bar{\mu}_{Rm}$		----------------
	Unbiased Measurement-Residual Computation	$\bar{Y}_{Dm}'=\bar{Y}_{Dm}+\sum_{s}\left[\bar{N}_{Dms}\dot{u}_{Ds}-\bar{M}_{Dms}\left(x_{Ds}+u_{Ds}\right)\right]$	$\bar{Y}_{Rm}'=\bar{Y}_{Rm}+\sum_{s}\left[\bar{M}_{Rms}\left(x_{Rs}+u_{Rs}\right)\right]$	----------------
		$\Delta\bar{Y}_m'=\bar{Y}_{Dm}'-\bar{Y}_{Rm}'$		
Once per Substate Execution	Partitioned Kalman Gains Computation	$\beta_{Dms}=\sum_{s'}\left(P_{Dss'}\bar{M}_{Dms'}^{T}-\bar{Z}_{Dms}^{T}\right)$	$\beta_{Rms}=\sum_{s'}\left(P_{Rss'}\bar{M}_{Rms'}^{T}\right)-\bar{Z}_{Rms}^{T}$	$\beta_{D/Rms}=\sum_{s'}\left(P_{D/Rss'}\bar{M}_{Rms'}^{T}\right)$ $\beta_{R/Dms}=\sum\left(P_{D/Rss'}^{T}\bar{M}_{Dms'}^{T}\right)$
		$b_{KDms}=\frac{1}{\bar{Q}}\left(\beta_{Dms}+\beta_{D/Rms}\right)$	$b_{KRms}=\frac{1}{\bar{Q}_m}\left(\beta_{Rms}+\beta_{R/Dms}\right)$	----------------
	Partitioned Nonrecursive Least Square Gains Comp'n	$b_{LDms}=\frac{1}{\bar{\mu}_m}\bar{M}_{Dms}^{T}$	$b_{LRms}=\frac{1}{\bar{\mu}_m}\bar{M}_{Rms}^{T}$	----------------
	Partitioned Overall Gains Computation	$\Delta b_{Dms}=b_{KDms}-b_{LDms}$	$\Delta b_{Rms}=b_{KRms}-b_{LRms}$	----------------
		$b_{Dms}=b_{KDms}-\zeta\Delta b_{Dms}$	$b_{RMs}=b_{KRms}-\zeta\Delta b_{Rms}$	----------------
	Estimate Substates Measurement Update Comp'n	$x_{Ds}=x_{Ds}+b_{Dms}\overline{\Delta Y}_m'$	$x_{Rs}=x_{Rs}+b_{Rms}\overline{\Delta Y}_m'$	----------------
Once per Substate Pair Execution	Covariance Submatrices Measurement Update Computation	$\Delta P_{Dmss'}=\bar{Q}_m\left(b_{KDms}b_{KDms'}^{T}-\zeta^2\Delta b_{Dms}\Delta b_{Dms'}^{T}\right)$	$\Delta P_{Rmss'}=\bar{Q}_m\left(b_{KRms}b_{KRms'}^{T}-\zeta^2\Delta b_{Rms}\Delta b_{Rms'}^{T}\right)$	$\Delta P_{D/Rmss'}=\bar{Q}_m\left(b_{KDms}b_{KDms'}^{T}-\zeta^2\Delta b_{Dms}\Delta b_{Rms'}^{T}\right)$
		$P_{Dss'}=P_{Dss'}-\Delta P_{Dmss'}$ **	$P_{Rss'}=P_{Rss'}-\Delta P_{Rmss'}$	$P_{D/Rss'}=P_{D/Rss'}-\Delta P_{D/Rmss'}$ **

*Depending on how much intermediate processing of the time averaged measurement set m1 is done prior to its use by this submodule (see Time Averaged Measurement Selection and Linear Combination submodules specifications).

**When ADR pseudostate DA2 is carried in x_D, P_D, and $P_{D/R}$, these operations must be followed by setting x_{D2}, P_{D2s}, and $P_{D/R2s}$, in accordance with the formulae footnoting the operations summary for the Estimate/Covariance Matrix Time Update submodule.

40989

TABLE XLVII. KFIM INPUT/OUTPUT SUMMARY

Inputs/Outputs \ Substate Class	D Substates	R Substates	D/R Substates
Inputs	$\underline{m = (m1)^{*}_{n-1}; s,s'=(sd')_{n-1}}$ $\left(x_{Ds}, P_{Dss'}\right)_{F,n-1}$ $\left(u_{Ds}\right)_{n-1}, \left(\dot{u}_{Ds}\right)_{n-2}$ $\left(\overline{Y}'_{Dm}, \overline{C}_{Dm}, \overline{M}_{Dms}, \overline{N}_{Dms}, \overline{Z}_{Dms}, \overline{W}_{Dms}\right)_{n-1}$	$\underline{m=(m1)^{*}_{n-1}; s,s'=(sr0)_{n-1}}$ $\left(x_{Rs}, P_{Rss'}\right)_{F,n-1}$ $\left(\dot{u}_{Rs}\right)_{n-2}$ $\left(\overline{Y}'_{Rm}, \overline{C}_{Rm}, \overline{M}_{Rms}, \overline{Z}_{Rms}, \overline{W}_{Rms}\right)_{n-1}$	$\underline{m=(m1)^{*}_{n-1}; s=(sd')_{n-1}; s'=(sr0)_{n-1}}$ $\left(P_{D/Rss'}\right)_{F,n-1}$
Outputs	$\underline{m=(m1)^{*}_{n-1}, s,s'=(sd')_{n}}$ $\left(x_{Ds}, P_{Dss'}\right)_{S,n}$ $\left(\overline{Q}_{Dm}, \overline{\mu}_{Dm}, \beta_{Dms}, b_{Kdms}, b_{LDms}, \Delta b_{Dms}, b_{Dms}, \Delta P_{Dmss'}\right)_{n}$	$\underline{m=(m1)^{*}_{n-1}; s,s'=(sr0)_{n}}$ $\left(x_{Rs}, P_{Rss'}\right)_{S,n}$ $\left(\overline{Q}_{Rm}, \overline{\mu}_{Rm}, \beta_{Rms}, b_{KRms}, b_{LRms}, \Delta b_{Rms}, b_{Rms}, \Delta P_{Rmss'}\right)_{n}$	$\underline{m=(m1)^{*}_{n}; s=(sd')_{n}; s'=(sr0)_{n}}$ $\left(\overline{\Delta Y}_{m}, \overline{Q}_{D/Rm}, \overline{Q}_{m}, \beta_{D/Rms}, \Delta P_{D/Rmss'}\right)_{n}$ $\left(P_{D/Rss'}\right)_{S,n}$

*Or m2 or m3.

III.4.f

KALMAN FILTER ESTIMATE/PROCESSOR CONTROL MODULE

(KCOM)

SPECIFICATION

This module first computes controls to be applied to processor variables associated with both non-Kalman (D and R) and Kalman modules, and then applies not only these, but a portion of the corresponding controls computed by this (KCOM) module in the last cycle.

In order to compute the non-Kalman module controls for application at the end of the current cycle (i.e., at time t_F), the current KF error estimate (i.e., the estimate just generated by current-cycle KFIM operations, which relates to processor error state at time t_S) is first predicted across the current cycle to time t_F.* Two types of KF estimate control -- impulsive and metered -- are then computed based on this t_F estimate, which can then be discarded. These estimate controls are then used to compute impulsive and metered control for the non-KF module variables which are then applied at time t_F.

On the other hand, the t_S estimate is retained, and is corrected for the non-KF module impulsive control applied at the end of the last cycle by this module (KCOM) by subtracting out the impulsive estimate control generated by this module in the last cycle. The corrected, retained t_S estimate thus corresponds to the processor error state at the beginning of the current cycle.**

*This is in fact a pure prediction based on at most the stationary and quasi-stationary portions of the time update matrices, since the non-stationary, vehicle-dynamics-dependent portions will not be available until completion of the (parallel) current-cycle KTMM operations.

**This estimate thus lags real time by one cycle, but includes all vehicle-dynamics-dependent error effects.

TABLE XLVIII. KCOM OPERATIONS SUMMARY (KF CONTROL)

Operation \ Substate Class			D Substates: IDR	D Substates: ADR	D Substates: PDR	R Substates
Execute Once per KF Cycle	Estimate Sync (Prediction Across Current Cycle)	Substate Index Satup	s,s',i,j = sd' (DR Mode Dependent)**			s,s'=sr0
		Estimate Substates Time Update (Across Current Cycle)	$(x_{Ds})_F = \sum_j \left[\phi_{Dsj}\left\{(x_{Dj})_S - u_{Dj}\right\} + G_{DSj}\dot{u}_{Dj}\right]$ (All s,j; $u_{Dj}, \dot{u}_{Dj}$ from prior cycle; rapid-dynamics-dependent ϕ_D and G_D submatrices null)**			$(x_{Rs})_F = \phi_{Rs}\left\{(x_{Rs})_S - u_{Rs}\right\}$ (All s; u_{Rs} from prior cycle; rapid-dynamics-dependent ϕ_R submatrices null)
	Estimate Control	Estimate Substates Impulsive Control Formulation	s=DI1,DI2, DI4-DI7	s=DA1,DA2, DA6,DA7	s=DP1-DP3	$u_{Rs} = (x_{Rs})_F$ (All s)
			$u_{Ds} = (x_{Ds})_F$			
			s=DI3	s=DA3	------------	
			$u_{Ds} = -\gamma_p = -(x_{Ds})_F$ $\delta\theta_p = T_{p/L}K_L T_{L/C}(x_{D1})_F$ $\phi_p = \phi_p + \delta\theta_p$			
			--------	s=DA4,DA5 $u_{Ds}=0$		
		Estimate Substates Metered Control Formulation	s=DI3 IF* $\dot{u}_{Ds} = -\omega_L$ or $-\omega_{Lmax}$ if $\omega_L <$ or $\geq \omega_{Lmax}$ IS $\dot{u}_{Ds} = 0$	s=DA1,DA2, DA4-DA7 $\dot{u}_{Ds}=0$	s=DP1-DP3 $\dot{u}_{Ds} = 0$	$\dot{u}_{Rs} = 0$ (All s)
			s=DI1-DI3, DI5-DI7 $\dot{u}_{Ds} = 0$	s=DA3 $\dot{u}_{Ds} = -$ (Function of x_{DA4}, x_{DA5})		
		Estimate Substate Control Execution	s=DI1-DI7	S=DA1-DA7	S=DP1-DP3	$(x_{Rs})_S = (x_{Rs})_S - u_{Rs}$ (u_{RS} from prior cycle)
			$(x_{Ds})_S = (x_{Ds})_S - u_{Ds}$ (u_{Ds} from prior cycle)			

$$I_1 = \begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix}$$

*This formulation applies to only that 3x1 portion of $\dot{u}_{D3}$ corresponding to the fixed bias error rate locations of the overall x_{D3} substate. ($\omega_L = \theta_p/\Delta t_K$)

**For ADR, additional DA2 pseudostate (x_{DA} only) updating required per the footnote in the operations summary for the estimate/covariance matrix time update submodule.

40990

TABLE XLIX. KCOM OPERATIONS SUMMARY (NON-KF MODULES CONTROL)

KF Cycle Endpoint Execution

DR Nav Modules

Module	Operation	IDR: IS	IDR: IF	ADR	PDR
VSTM	Position Correction	$p = p - u_{D1}$* (IPA1)			
	Velocity Correction	$v = v - u_{D2}$ (IPA2)			
	Pseudo Spec Force Correction	--------------			$\beta_L = \beta_L - u_{DP3}$ (P1)
PLAM	$T_{P/L}$ Transformation Correction	$T_{L/C} = Q(-\phi_P)T_{P/L}$ (IA1)			
	$T_{L/C}$ Transformation Correction	$T_{L/C} = \{Q(-\phi_P)T_{P/L}\}^T Q(+\psi_P)T_{P/C}$ (IA2)			
	Δf_K Correction	(IS1)[a]	(IF1)[a]	------	
	$\Delta\omega_K$ Correction	(IS2)[b]	(IF2)[b]	------	
	$(\omega_{P/L})_P$ Correction	------	$\omega_K = \dot{u}_{DI3}$ (IF3)	$\omega_K = \dot{u}_{DA3}$ (AI)	
WASM	Wind Estimate Correction	------------		$(v_W)_L = (v_W)_L - u_{DA6}$ (A2)	
	TAS/TAS Transformation Corrections	------------		(A3)[c]	

Ref Nav Modules

Module	Operation	
ALTM	Reference Altitude Correction	$\Delta h_{BK} = \Delta h_{BK} - u_{RuA}$ (AB) $h_{PC} = h_{PC} - u_{RuA}$ (APS)
TDPM	User Clock Correction	$\begin{bmatrix} \Delta_K t_u \\ \Delta_K \dot{t}_u \end{bmatrix} = \begin{bmatrix} \Delta_K t_u \\ \Delta_K \dot{t}_u \end{bmatrix} - u_{RuC}$
	nth Emitter Net Clock Correction	$\begin{bmatrix} \Delta_K t_n \\ \Delta_K \dot{t}_n \end{bmatrix} = \begin{bmatrix} \Delta_K t_n \\ \Delta_K \dot{t}_n \end{bmatrix} - u_{REnC}$
	nth Emitter Net Position/ Velocity Correction	Function of u_{REnE}; depends on (a) emitter net type, (b) TWEM ephemeris computations type, and (c) KF error model selected
	jth Emitter nth Net kth Signal Propagation Error Compensation	Function of u_{REnjk}; depends on (a) emitter net type, (b) TPCM propagation compensation type, and (c) KF error model selected

*All $u_{Ds}, \dot{u}_{Ds}, u_{Rs}, \dot{u}_{Rs}$ indicated in this table are current cycle values (i.e., values determined by estimate substate control formulation computations in the current cycle)

Note: $Q(a) = I \cos|a| + (1-\cos|a|) \frac{a}{|a|}\frac{a^T}{|a|} \cdot \sin|a| \frac{ax}{a}$

(a = 3x1 vector)

a: Function of u_{DI6}, u_{DI7}; depends on PLAM Δf compensation formulation

b: Function of u_{DI4} and u_{DI5}; depends on PLAM $\Delta\omega$ compensation formulation

c: Function of u_{DA7}; TBD

40991

TABLE L. KCOM INPUT/OUTPUT SUMMARY (KF CONTROL)

Substate Class / Inputs/Outputs	D Substates	R Substates
Inputs	$\underline{s=(sd')_n}$ $(x_{Ds})_{S,n}, (\phi_{Dss'}, G_{Dss'})^{**}_{(F,S)n}$ $\underline{s=(DA2)^*_n}$ $(H_{vAs})_{S,n}$ $\underline{s=(sd')_{n-1}}$ $(u_{Ds}, \dot{u}_{Ds})_{n-1}$	$\underline{s=(sr0)_n}$ $(x_{Rs})_{S,n}, (\phi_{Rs})^{**}_{(F,S)n}$ $\underline{s=(sr0)_{n-1}}$ $(u_{Rs})_{n-1}$
Outputs	$\underline{s=(sd')_n}$ $(x_{Ds})_{F,n}, (x_{Ds})_{S,n},$ $(u_{Ds}, \dot{u}_{Ds})_n$ $\underline{s=(DA2)^*_n}$ $(x_{Ds})_{F,n}$	$\underline{s=(sr0)_n}$ $(x_{Rs})_{F,n}, (x_{Rs})_{S,n},$ $(u_{RS})_n$

*Additional Input/output Requirements for ADR Only
**Rapid-dynamics-dependent ϕ and G Submatrices Null

TABLE LI. KCOM INPUT/OUTPUT SUMMARY (NON-KF MODULES CONTROL)

Module Computation Class / Input/Output	D MODULES	R MODULES
Input	<u>VSTM</u>: p, v, u_{D1}, u_{D2} (IPA) β_L, u_{DP3} (P) <u>PLAM</u>: $T_{P/L}, T_{P/C}, u_{D3} = \psi_p, \phi_p$ (IA) $u_{DI4} - u_{DI7}$ (I) $\dot{u}_{DI3}$ (IF), $\dot{u}_{DA3}$ (A) <u>WASM</u>: $(V_W)_L, u_{DA6}, u_{DA7}$ (A)	<u>ALTM</u>: u_{RUA} (A), Δh_{BK} (ΔB), h_{pc} (APS) <u>TDPM</u>: u_{RUC}, u_{RE}'S $+ \Delta_K t_U$, $\Delta_K e_j$, $\Delta_K \dot{t}_U$, $\Delta_K \dot{e}_j$, $\Delta_K t_n$, $\Delta_K L_j$ $\Delta_K \dot{t}_n$,
Output	<u>VSTM</u>: p, v (IPA) β_L (P) <u>PLAM</u>: $T_{P/C}, T_{L/C}, \omega_K$ (IA) $\Delta f_K, \Delta\omega_K$ (I) <u>WASM</u>: $(v_W)_L$ + TBD (A)	<u>ALTM</u>: Δh_{BK} (AB), h_{PC} (APS) <u>TDPM</u>: $\Delta_K t_U$, $\Delta_K e_j$, $\Delta_K \dot{t}_U$, $\Delta_K \dot{e}_j$, $\Delta_K t_n$, $\Delta_K L_j$ $\Delta_K \dot{t}_n$,

III.4.g

KALMAN FILTER TIME UPDATE
MATRIX GENERATION MODULE

(KTMM)

SPECIFICATION

This module generates the time update matrices for use by the KTUM on the next cycle. The algorithms are shown in their fundamental differential equation form, since either recursive, or single-pass closed form, or a mixture of both of these types of solution may be required and/or desirable, depending on the nature of the application.

TABLE LII. D SUBSTATES KTMM OPERATIONS SUMMARY (GOVERNING DIFFERENTIAL EQUATIONS)

Operation \ DR Mode			IDR	PDR	ADR
Execution Once Per KF Cycle	D Substate Index Setup		s,s'=sd'=DI1+DI2, DI3-DI7	s,s'=sd'= DP1+DP2 DP3	s,s'= DA1+DA3, DA4-DA7
	D Substate Prediction Submatrices Initialisation		$\Phi_{Dss}=I$ (All s) $\Phi_{Dss'}=0$ (All $s \neq s'$) $R_{Dss'}=0$ (All ss') $G_{Dss'}=0$		
Execution: Depends Both on Error Dynamics Variability and Measurement Data Rate Relative to KF Cycle Rate	D Substate Prediction Submatrices Generation	Principal Autonomous Sub-matrices	<u>All s</u> $\dot{\Phi}_{Dss}=A_{Dss}\Phi_{Dss}$ <u>s=DI4</u> $\dot{G}_{Dss}=I+A_{Dss}G_{Dss}$ <u>s=DI4 thru DI7</u> $\dot{R}_{Dss}=K_{Ds}$ $+A_{Dss}R_{Dss}+(A_{Dss}R_{Dss})^T$	<u>All s</u> $\dot{\Phi}_{Dss}=A_{Dss}\Phi_{Dss}$ <u>s=DP3</u> $\dot{R}_{Dss}=K_{Ds}$ $+A_{Dss}R_{Dss}+(A_{Dss}R_{Dss})^T$	<u>All s</u> $\dot{\Phi}_{Dss}=A_{Dss}\Phi_{Dss}$ <u>s=DA4 thru DA7</u> $\dot{R}_{Dss}=K_{Ds}$ $+A_{Dss}R_{Dss}+(A_{Dss}R_{Dss})^T$
		Principal Substate Coupling Sub-matrices	<u>s=DI1+DI2, s'=DI3, DI6, DI7; s=DI3, s'=DI4, DI5</u> $\dot{\Phi}_{Dss'}=A_{Dss}\Phi_{Dss'}$ $+A_{Dss'}\Phi_{Ds's'}$ $\dot{R}_{Dss'}=$ $A_{Dss}R_{Dss'}+A_{Dss'}R_{Ds's'}$ +(")^T <u>s=DI3, s'=DI4</u> $\dot{G}_{Dss'}=A_{Dss}G_{Dss'}+A_{Dss'}G_{Ds's'}$	<u>s=DP1+DP2, s'=DP3</u> $\dot{\Phi}_{Dss'}=A_{Dss}\Phi_{Dss'}$ $+A_{Dss'}\Phi_{Dss'}$ $\dot{R}_{Dss'}=$ $A_{Dss}R_{Dss'}+A_{Dss'}R_{Ds's'}$ +(")^T	<u>s=DA1+DA3, s'=DA4 thru DA7</u> $\dot{\Phi}_{Dss'}=A_{Dss}\Phi_{Dss'}$ $+A_{Dss'}\Phi_{Ds's'}$ $\dot{R}_{Dss'}=$ $A_{Dss}R_{Dss'}+A_{Dss'}R_{Ds's'}$ +(")^T
		Secondary Substate Coupling Sub-matrices	<u>s=DI1+DI2, s'=DI3, s"=DI4, DI5</u> $\dot{\Phi}_{Dss''}=A_{Dss}\Phi_{Dss''}+A_{Dss'}\Phi_{Ds's''}$ $\dot{R}_{Dss''}=$* <u>s=DI1+2, s'=DI3, s"=DI4</u> $\dot{G}_{Dss''}=A_{Dss}G_{Dss''}+A_{Dss'}G_{Ds's''}$	. . . .	. . . .

*To conserve space, the IDR Secondary Coupling and Tertiary Autonomous R Submatrix equations are not shown here. If needed, these are also obtainable from the full D State equation, $\dot{R}_D = K_D + A_D R_D + R_D A_D^T$.

TABLE LIII. R SUBSTATES KTMM OPERATIONS SUMMARY (GOVERNING DIFFERENTIAL EQUATIONS)

<table>
<tr><th colspan="3">Operation</th><th></th></tr>
<tr><td rowspan="2">Execution Once Per KF Cycle</td><td colspan="2">R Substate Index Setup</td><td>$s = sro$</td></tr>
<tr><td colspan="2">R Substate Prediction Submatrices Initialization</td><td>$\phi_{Rs} = I$
$R_{Rs} = 0$</td></tr>
<tr><td rowspan="2">Execution*</td><td rowspan="2">R Substate Prediction Submatrices Generation</td><td>Principal Autonomous Sub-matrices</td><td>$\dot{\phi}_{Rs} = A_{Rs}\,\phi_{Rs}$
$\dot{R}_{Rs} = K_{Rs} + A_{Rs}R_{Rs} + (A_{Rs}R_{Rs})^T$</td></tr>
<tr><td>Other Sub-matrices</td><td>------------</td></tr>
</table>

*<u>Execution</u>:

- Once per KF cycle if $A_{Rs} = \text{constant}$
- Based on $\int_{\Delta t} A_{Rs}\,dt$ if $A_{Rs} \neq \text{constant}$

TABLE LIV. KTMM INPUT/OUTPUT SUMMARY

Inputs/Outputs \ Substate Class	D Substates	R Substates
Inputs	s = sd' $(A_{Dss'}, K_{Ds})_{i,n}$	s = sro $(A_{Rs}, K_{Rs})_{i,n}$
Outputs	s = sd' $(\phi_{Dss'}, G_{Dss'}, R_{Dss'})_{(F,S)n}$ $(\phi_{Dss'}, G_{Dss'}, R_{Dss'})_{i,n}$	s = sro $(\phi_{Rs}, R_{Rs})_{(F,S)n}$ $(\phi_{Rs}, R_{Rs})_{i,n}$

III.4.h

KALMAN FILTER MEASUREMENT MATRIX GENERATION MODULE

(KMMM)

SPECIFICATION

This module KF endpoint-synchronizes and time-smooths current-cycle raw D/R synchronous measurement-differences and their associated measurement matrices. Its outputs are further processed by the measurement preprocessing matrices (KMRM, KMCM, KMOM) in the next KF cycle. The algorithms are shown in their fundamental, nonrecursive summation forms. Compact closed-form, or recursive, or a mixture of both types (all based on these fundamental forms) may be required and/or desirable, depending on the nature of the application.

TABLE LV. KMMM OPERATIONS SUMMARY

Operation (rows) / Substate Class (columns)		D Substates	R Substates
Execute Once per KP Cycle	Input Measurement Type Index Setup	$m = m1$	
	$\bar{Y}_m$, $\bar{C}_m$ Computations	$\bar{Y}_{Dm}=\frac{1}{n}\sum_i Y_{Dmi}$, $\bar{C}_{Dm}=\frac{1}{n}\sum_i C_{Dmi}$ (All m)	$\bar{Y}_{Rm}=\frac{1}{n}\sum_i Y_{Rmi}$, $\bar{C}_{Rm}=\frac{1}{n}\sum_i C_{Rmi}$ (All m)
	ΔY_m, $\overline{\Delta Y}_m$ Computations	$\Delta Y_{mi}=Y_{Dmi}-Y_{Rmi}$ (All m) $\overline{\Delta Y}_m=\bar{Y}_{Dm}-\bar{Y}_{Rm}=\frac{1}{n}\sum_i \Delta Y_{mi}$	
Execute Once per KP Cycle for Each Measurement of Subset M1	$\bar{M}_{ms}$ Computations	<u>IDR</u> <u>m=1-5:</u> s'=DI1+DI2, s=DI1+DI2, DI3-DI7 <u>PDR</u> <u>m=1-5:</u> s'=DP1+DP2, s=DP1+DP2, DP3 <u>ADR</u> <u>m=1-3,5:</u> s'=DA1+DA3, s=DA1+DA3, DA4-DA7 <u>m=4:</u> s'=DA1+DA3, s=DA1+DA3, DA4,DA5 $\bar{M}_{Dms}=\frac{1}{n}\sum_i M_{Dms'i}\Phi_{Ds'si,F}$ <u>IDR</u> <u>m=1-5:</u> s'=DI1+DI2, s=DI4 $\bar{N}_{Dms}=\frac{1}{r}\sum_i M_{Dms'i}G_{Ds'si,F}$ <u>ADR</u> <u>m=4:</u> s'=DA1+DA3, s=DA6,DA7 $\bar{M}_{Dms}=\frac{1}{n}\sum\left(M_{Dms'i}\Phi_{Ds'si,F}+M_{Dmsi}\Phi_{Dssi,F}\right)$	m=2: s=RUA <u>m=3-5:</u> s=RUC,REnC, <u>REnE</u>,REnjk $\bar{M}_{Rms}=\frac{1}{n}\sum_i M_{Rmsi}\Phi_{Rsi,F}$
	$\bar{Z}_{ms}$, $\bar{W}_{ms}$ Computations	**	$\bar{Z}_{Rms}=\frac{1}{n}\sum M_{Rmsi}\Phi_{Rsi,F}R_{sF,i}$ $W_{Rms}=\frac{1}{n^2}\sum_i\sum_j M_{Rmsi,ij^*}R_{RsF,ij^*}M^T_{Rmsj,F}$

ij* = Larger of i and j.

**Omitted here for lack of space. See Appendix VIII

40993

TABLE LVI. KMMM INPUT/OUTPUT SUMMARY

Inputs/Outputs \ Substate Class	D Substates	R Substates	D/R Substates
Inputs	$\underline{m=(ml)_n;s,s'=(sd')_n}$ $\left(Y_{Dm}, C_{Dm}, M_{Dms}, \phi_{Dss'}, G_{Dss'}, R_{Dss'}\right)_{i,n}$	$\underline{m=(ml)_n, s=(sr0)_n}$ $\left(Y_{Rm}, C_{Rm}, M_{Rms}, \phi_{Rs}\right)_{i,n}$	
Outputs	$\underline{m=(ml)_n;s,s'=(sd')_n}$ $\left(\overline{Y}_{Dm}, \overline{C}_{Dm}, \overline{M}_{Dms}, \overline{N}_{Dms}, \overline{Z}_{Dms}, \overline{W}_{Dms}\right)_n$	$\underline{m=(ml)_n,(s=sr0)_n}$ $\left(\overline{Y}_{Rm}, \overline{C}_{Rm}, \overline{M}_{Rms}, \overline{Z}_{Rms}, \overline{W}_{Rms}\right)_n$	$\underline{m=(ml)_n}$ $\left(\Delta Y_m\right)_{i,n}$ $\left(\overline{\Delta Y}_m\right)_n$

III.5

INITIALIZATION AND SWITCHING MODULES

SPECIFICATIONS

These modules together comprise the operations necessary to initiate and subsequently to switch processor navigation with regard to DR mode, R configuration, and computational reference (C) frame.

III.5.a

NAVIGATION START-UP MODULE

(NSTM)

SPECIFICATION

This module initializes processor dynamic operation before first entry to the main, dynamic navigation loop for PDR operation.

Specifically, it initializes (a) the computational frame (to C = E); (b) initial position, velocity, and pseudo-acceleration to null vectors; and (c) the KF PDR error substate and control vectors to null, and the covariance matrix to large values to reflect the consequent large uncorrelated uncertainties in the initial position, velocity and the pseudo-acceleration error substates.

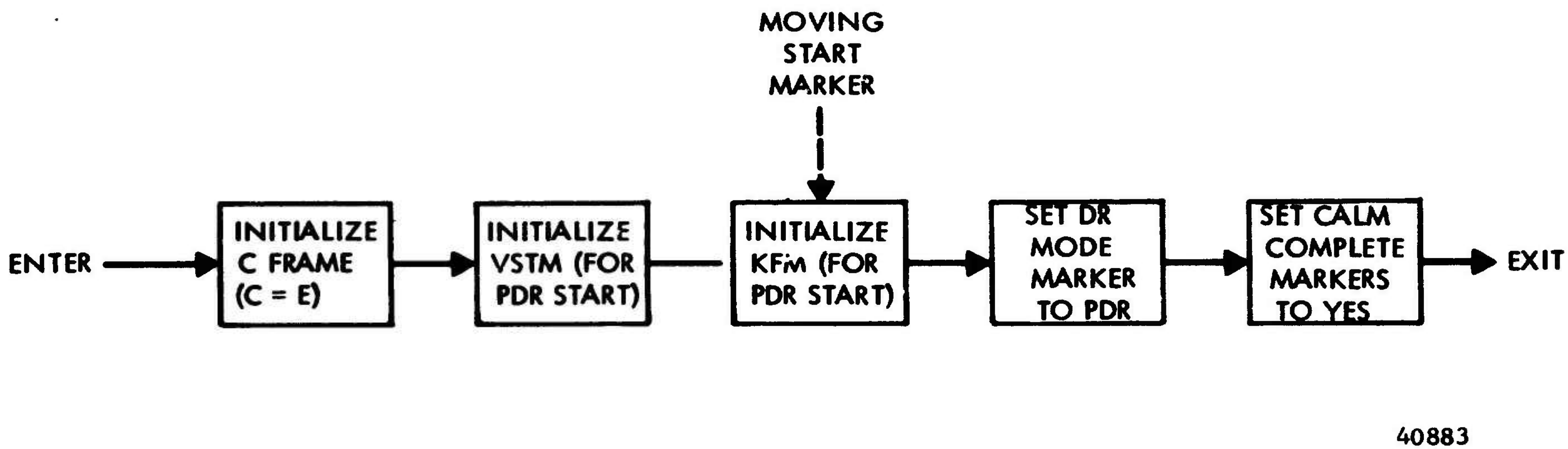


Figure 22. NSTM Operations Flow

TABLE LVII. NSTM OPERATIONS SUMMARY

<table>
<tr><th colspan="2">Start Type / Operation</th><th>Fixed Start</th><th>Moving Start</th></tr>
<tr><td rowspan="5">Execution Prior to Dynamic Nav Loop Entry*</td><td>C Frame Initialization (C = E)</td><td colspan="2">$C_{C/E} = 0, \; T_{C/D} = I$</td></tr>
<tr><td>VSTM Initialization (for PDR Start)</td><td colspan="2">$p = 0 \quad \beta_L = 0$
$v = 0 \quad \omega_{E/I} = \text{Constant}$</td></tr>
<tr><td rowspan="2">KFM/DR Initialization (for PDR Start)</td><td colspan="2"><u>s,s' = DP1-DP3</u>
$x_{Ds} = 0 \quad P_{Dss'} = 0$
$u_{Ds} = 0 \quad P_{Dss'} = \sigma_{PO}^2 I$ (s = DP1)
$\dot{u}_{Ds} = 0 \quad \sigma_{PO} = \frac{\sqrt{3}}{3} R_0$; R_0 = nom. earth radius)</td></tr>
<tr><td>--------------------</td><td><u>s=DP2</u>
$P_{Dss} = \sigma_{v0}^2 I$ ($\sigma_{v0} = \frac{\sqrt{3}}{3} |v_{CR}|$;
$|v|_{CR}$ = nom. vehicle cruise speed)
<u>s=DP3</u>
$P_{Dss} = \begin{bmatrix} \sigma_{\beta 1}^2 & 0 & 0 \\ 0 & \sigma_{\beta 2}^2 & 0 \\ 0 & 0 & \sigma_{\beta 3}^2 \end{bmatrix}$</td></tr>
<tr><td></td><td colspan="2"></td></tr>
</table>

*Execute Operations in Order Shown

TABLE LVIII. NSTM INPUT/OUTPUT SUMMARY

Start Type / Variable Type	Fixed or Moving	Moving Only
Input	σ_{p0}, Earth Rate	$\sigma_{v0}, \sigma_{\beta i}$ (i=1,2,3)
Output	<u>Init Values for:</u> $T_{C/E}, \tilde{C}_{C/E}, p, v, \omega_{E/I}, \beta_L$ $\left(x_{Ds}, u_{Ds}, \dot{u}_{Ds}, P_{Dss'}\right)_{s,s'=DP1-DP3}$	<u>Init Values for:</u> P_{Dss} (s=DP1,DP2)

[redacted]

III.5.b

C FRAME SWITCHING MODULE (NON-KF MODULES)

(CSWM)

SPECIFICATION

This module embeds the operations necessary to (a) define a new computational reference frame, (b) generate the switching transformation and center-displacement vector between the old and new computational frames, and then (c) switch all affected processor D and R variables to the new frame. Actual switching of the variables is delayed until the end of the KF cycle in which the new C frame command was initiated, at which time affected K module variables are also switched by (a submodule of) the KSWM.

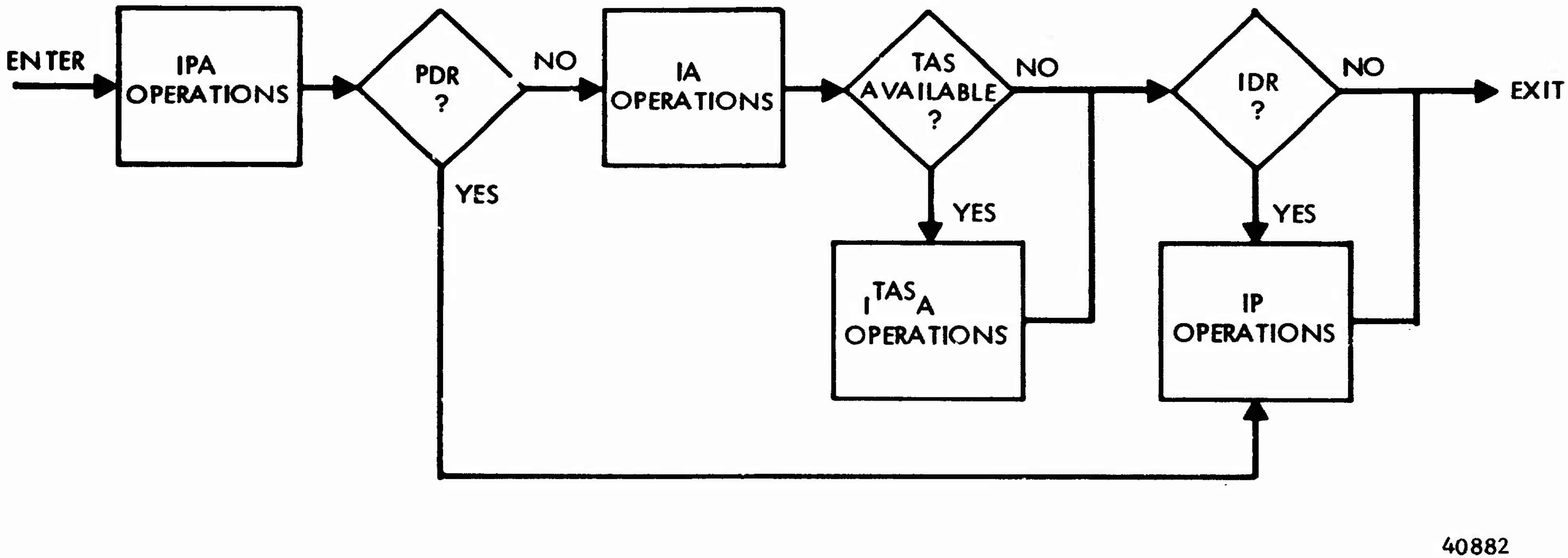


Figure 23. CSWM Operations Flow (Non-KF Modules)

TABLE LIX. CSWM OPERATIONS SUMMARY (NON-KF MODULES)

<table>
<tr><th colspan="3" rowspan="2">DR Nav Mode / Operation</th><th colspan="2">IDR(I)</th><th rowspan="2">ADR(A)</th><th rowspan="2">PDR(P)</th></tr>
<tr><th>Without TAS</th><th>With TAS</th></tr>
<tr><td rowspan="13">nth KF Cycle Endpoint Execution*</td><td colspan="2">New C Frame Definition $T_{C/E}, C_{C/E}$</td><td colspan="4">$\left(C_{C/E}\right)_{NEW}$ = Function of New C Frame Definition Data (IPA1)
$\left(T_{C/E}\right)_{NEW}$ = Function of New C Frame Definition Data</td></tr>
<tr><td colspan="2">Old-to-New C Frame Rotation Matrix T_{SW}</td><td colspan="4">$T_{SW} = \left(T_{C/E}\right)_{NEW} \left(T_{C/E}\right)^{T}_{OLD}$ (IPA2)</td></tr>
<tr><td colspan="2">Old-to-New C Frame Center Translation Vector ΔC_{SW}</td><td colspan="4">$\Delta C_{SW} = \left(C_{C/E}\right)_{NEW} - \left(C_{C/E}\right)_{OLD}$ (IPA3)</td></tr>
<tr><td colspan="2">Old Definition Replacement by New</td><td colspan="4">$\left(C_{C/E}\right)_{NEW} \rightarrow \left(C_{C/E}\right)_{OLD}$ $\left(T_{C/E}\right)_{NEW} \rightarrow \left(T_{C/E}\right)_{OLD}$ (IPA4)</td></tr>
<tr><td rowspan="9">Navigation Vectors/Matrices Switching</td><td rowspan="6">DR Nav Modules</td><td colspan="4">$a_{NEW} = T_{SW} a_{OLD} - \left(T_{C/E}\right)_{NEW} \Delta C_{SW}$ $(a = p)$ (IPA5)</td></tr>
<tr><td colspan="4">$a_{NEW} = T_{SW} a_{OLD}$ $\left(a = v, g, \Delta p, T_{L/C}\right)$ (IPA6)</td></tr>
<tr><td colspan="3">$a_{NEW} = T_{SW} a_{OLD}$ $\left(a = T_{P/C}, T_{A/C}, \omega_{A/C}, \omega_{E/I}\right)$ (IA1)</td><td>$a_{NEW} = T_{SW} a_{OLD} \left(a = u_1, u_2, u_3\right)$ (P1)</td></tr>
<tr><td>-----------</td><td colspan="2">$a_{NEW} = T_{SW} a_{OLD} \left(a = v_W, v_{AS}\right)$ (I(TAS) A)</td><td>--------------------------</td></tr>
<tr><td colspan="2">$a_{NEW} = T_{SW} a_{OLD} (a = f, \Delta v)$ (IP1)</td><td>-------------</td><td>(IP1)</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td rowspan="3">Ref Nav Modules</td><td colspan="4">$a_{NEW} = T_{SW} a_{OLD} - \left(T_{C/E}\right)_{NEW} \Delta C_{SW}$ $\left(a = e_{nj}\right)$ (All nj in sr0) (IPA7)</td></tr>
<tr><td colspan="4">$a_{NEW} = T_{SW} a_{OLD}$ $\left(a = \dot{e}_{nj}\right)$ (All nj in sr0) (IPA8)</td></tr>
<tr><td colspan="4"></td></tr>
</table>

*New C frame commanded by operator in nth KF cycle.

40994

TABLE LX. CSWM INPUT/OUTPUT SUMMARY (NON-KF MODULES)

Inputs/Outputs		DR Mode Subsets	IPA	IA	$I^{TAS}{}_A$	IP	P
Inputs	Dynamic	DR Nav Modules	$(p,v,g,\Delta p,T_{L/C})_{OLD}$	$(T_{P/C},T_{A/C},\omega_{A/C})_{OLD}$ $(\omega_{E/I})_{OLD}$	$(v_W,v_{AS})_{OLD}$	$(\Delta v,f)_{OLD}$	$(u_1,u_2,u_3)_{OLD}$
		Ref Nav Modules	$(e_{nj},\dot{e}_{nj})_{OLD}$ (All nj in sr0)				
	Constant	DR Nav Modules	New C Frame Def. Data $(T_{C/E},C_{C/E})_{OLD}$				
		Ref Nav Modules					
Outputs	Dynamic	DR Nav Modules	$(p,v,g,\Delta p,T_{L/C})_{NEW}$	$(T_{P/C},T_{A/C},\omega_{A/C})_{NEW}$ $(\omega_{E/I})_{NEW}$	$(v_W,v_{AS})_{NEW}$	$(\Delta v,f)_{NEW}$	$(u_1,u_2,u_3)_{NEW}$
		Ref Nav Modules	$(e_{nj},\dot{e}_{nj})_{NEW}$ (All nk in sr0)				
	Constant	DR Nav Modules	$(C_{C/E},T_{C/E})_{NEW}$, $T_{SW},\angle C_{SW}$				
		Ref Nav Modules					

III.5.c

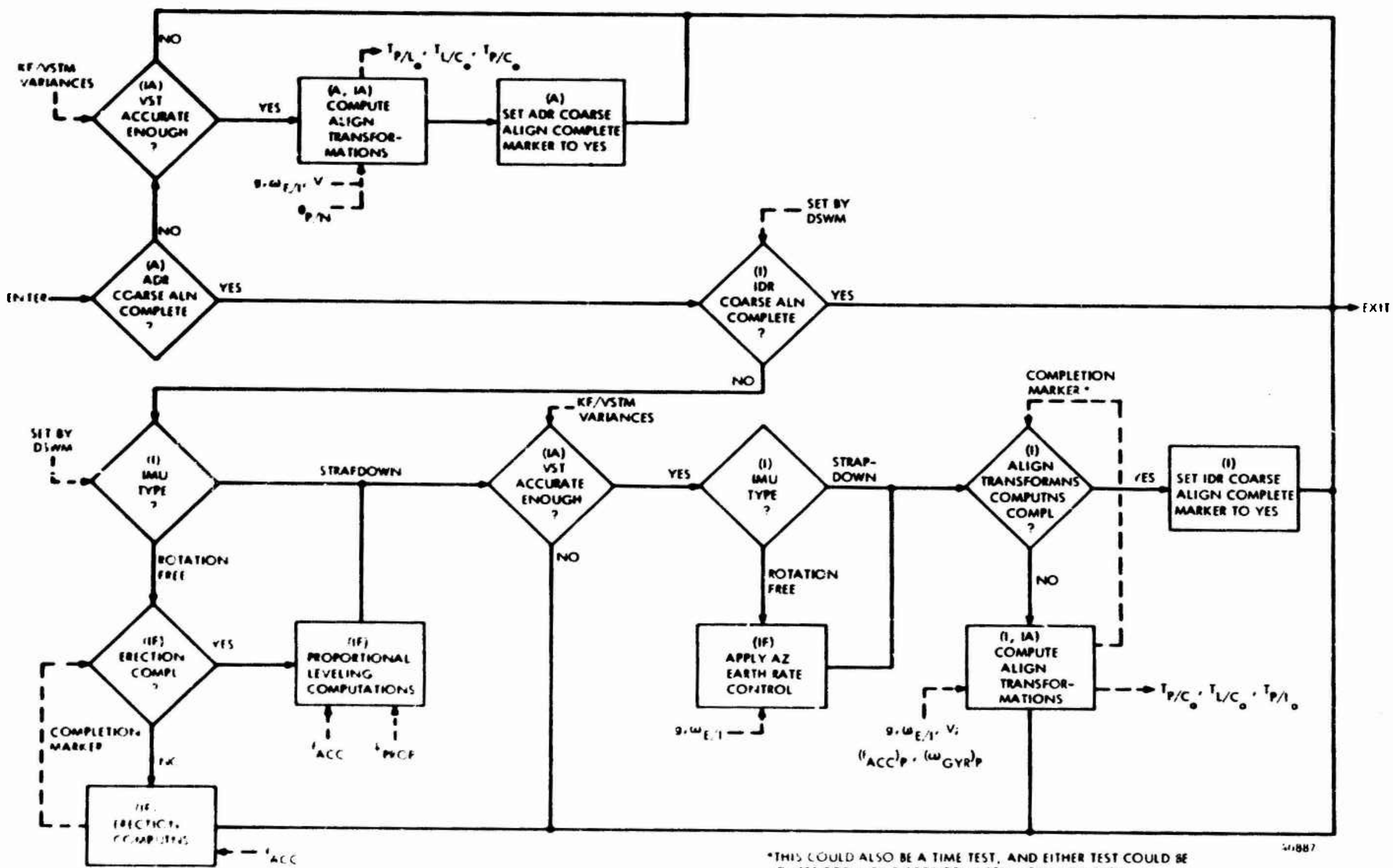
COARSE ALIGN MODULE

(CALM)

SPECIFICATION

This module is used to accomplish coarse (IMU or AHRU) platform-to-computer alignment prior to IDR or ADR operation.

In particular, initial IMU operations consist of coarse-leveling the platform (if it is of the rotationally isolated type). When this is complete, no further operations are initiated unless VSTM position and velocity data is sufficiently accurate (as indicated by the corresponding KF variances). When this latter condition is met, coarse computational alignment -- i.e., coarse determination of the (AHRU or IMU) platform-to-computer and L-frame-to-computer transformations for subsequent PLAM IDR or ADR navigation updating -- begins and continues until the initiation of the IDR or ADR navigation.



*THIS COULD ALSO BE A TIME TEST, AND EITHER TEST COULD BE OVERRIDDEN BY DIRECT OPERATOR MODE CONTROL.

Figure 24. CALM Operations Flow

TABLE LXI. CALM OPERATIONS SUMMARY

<table>
<tr><th rowspan="2" colspan="2">Align For / Operation</th><th colspan="2">IDR(I)</th><th rowspan="2">ADR(A)*</th></tr>
<tr><th>Strapdown (IS)</th><th>Rotation Free (IF)</th></tr>
<tr><td rowspan="4">Erection Phase</td><td>VST Accuracy Test</td><td colspan="3"><u>Pass</u>: Trace $P_{D11} \le \sigma_{PD}^2$ and Trace $P_{D22} \le \sigma_{v0}^2$ (IA1)
<u>Fail</u>: Either $P_{D11} > \sigma_{P0}^2$ or $P_{D22} > \sigma_{v0}^2$</td></tr>
<tr><td></td><td colspan="3"></td></tr>
<tr><td>Coarse Erection</td><td></td><td>$(\ell_1)_P = (f_{ACC})_P / \lvert f_{ACC} \rvert$ (IF1)
$(\omega_{SLEW})_P = f\left\{(\ell_1)_P, \lvert \omega_{SLEW} \rvert, \Delta\theta_L\right\}$</td><td></td></tr>
<tr><td>Proportional Leveling</td><td></td><td>$(u_1)_P = (P_1)_P \times (\ell_1)_P$ (IF2)
$(\omega_{PROPL})_P = -k_{PROPL} u_{1P}$</td><td></td></tr>
<tr><td rowspan="5">Alignment Phase</td><td>Azimuth Rate Control</td><td></td><td>$(\omega_{PROP1})_P = \left\{(\omega_{E/I})^T \frac{g}{\lvert g \rvert}\right\}(P_1)_P$ (IF3)</td><td></td></tr>
<tr><td>Total Gyro Rate</td><td>$(\omega_{GYR})_P$ = Strapdown Gyro Outputs</td><td>$(\omega_{GYR})_P = (\omega_{PROPL})_P + (\omega_{PROP1})_P$ (IF4)</td><td></td></tr>
<tr><td rowspan="3">Alignment Transformations</td><td colspan="2">$T_{P/Ci} = Q_{Pi} Q_{Ci}^{-1}$
$Q_{Pi} = \left[(f_{ACC})_P \,\vdots\, (\omega_{GYR})_P \,\vdots\, (f_{ACC})_P \times (\omega_{GYR})_P\right]_i$
$Q_{Ci} = \left[-g \,\vdots\, \omega_{P/I} \,\vdots\, -g \times \omega_{P/I}\right]_i$ (I1)
$\bar{T}_{P/C1} = T_{P/C1}$
$(i = 2,3,\ldots,n)$
$\bar{T}_{P/Ci} = (1-k_s)\bar{T}_{P/Ci-1} + k_s T_{P/Ci}$
$T_{R/CO} = \bar{T}_{P/Cn}$</td><td>$T_{P/CO} = T_{P/LO} T_{L/CO}$ (A1)</td></tr>
<tr><td colspan="3">$\omega_{P/I} = \omega_{E/I} - \frac{1}{R0} \frac{g}{g} \times v$
$T_{CLO} = \left[(L_1)_C \,\vdots\, (L_2)_C \,\vdots\, (L_3)_C\right]$ (IA2)
$(L_1)_C = -\frac{g}{\lvert g \rvert}$ $(L_2)_C = \frac{(\omega_{P/I}) \times (L1)_C}{\lvert \omega_{P/I} \times (L1)_C \rvert}$ $(L_3)_C = (L_1)_C \times (L_2)_C$
$T_{L/CO} = T_{C/LO}^T$</td></tr>
<tr><td colspan="2">$T_{P/LO} = T_{P/CO} T_{C/LO}$ (I2)</td><td>$T_{P/LO} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{P/L} & -\sin\theta_{P/L} \\ 0 & \sin\theta_{P/L} & \cos\theta_{P/L} \end{bmatrix}$
$\theta_{P/L} = \theta_{P/N} + \theta_{N/L}$ (A2)
$\sin\theta_{N/L} = (L2)_C^T (L2)_{CO}$
$\cos\theta_{N/L} = (L2)_C^T (L3)_{CO}$
$(L2)_{CO} = (L2)_C$ with v set to 0
$(L3)_{CO} = (L1)_C \times (L2)_{CO}$</td></tr>
</table>

*Single fast-loop (MODM) cycle execution of CALM ADR equations is assumed

40995

TABLE LXII. CALM INPUT/OUTPUT SUMMARY

Variable Type \ Align For	I	IF	A	IA
Input	f_{ACC}, ω_{GYR}, k_s, n	$\Delta\theta_L, \lvert\omega_{SLEW}\rvert$, k_{PROPL}, k_{PROP1}	$\theta_{P/N}$	P_{D11}, P_{D22}, $\sigma^2_{P0}, \sigma^2_{v0}$, $g, \omega_{E/I}, v$, R_0
Output	-----	$(\ell_1)_P$, $(\omega_{SLEW})_P$, $(\omega_{PROPL})_P$, $(\omega_{PROP1})_P$	-----	$T_{P/CO}, T_{L/CO}$, $T_{P/LO}$

III.5.d

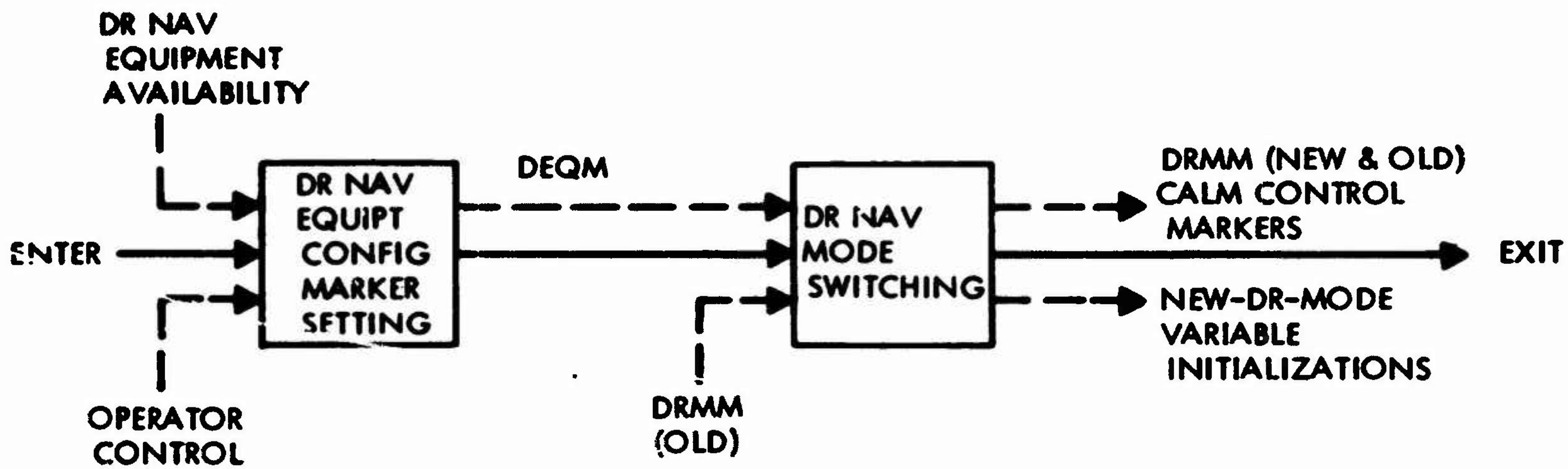
DR NAV MODE SWITCHING MODULE

(DSWM)

SPECIFICATION

This module (a) selects the appropriate (i.e., highest-level) DR mode of operation based on current DR equipment navigation output data availability, (b) sets the corresponding DR navigation mode marker (DRMM) which controls the DR configuration of the D modules and the CSWM (and is also used by the KSWM), and (c) if the mode is a new one, executes the required set-up for that mode.

In particular, if the new mode selected involves a platform, PDR operation is continued until CALM-controlled coarse alignment is complete. Also, the additional new-mode set-up specified is minimal because, and only if, the first new-mode cycle execution order of each of the D modules is that shown in the operations summary table for that module.

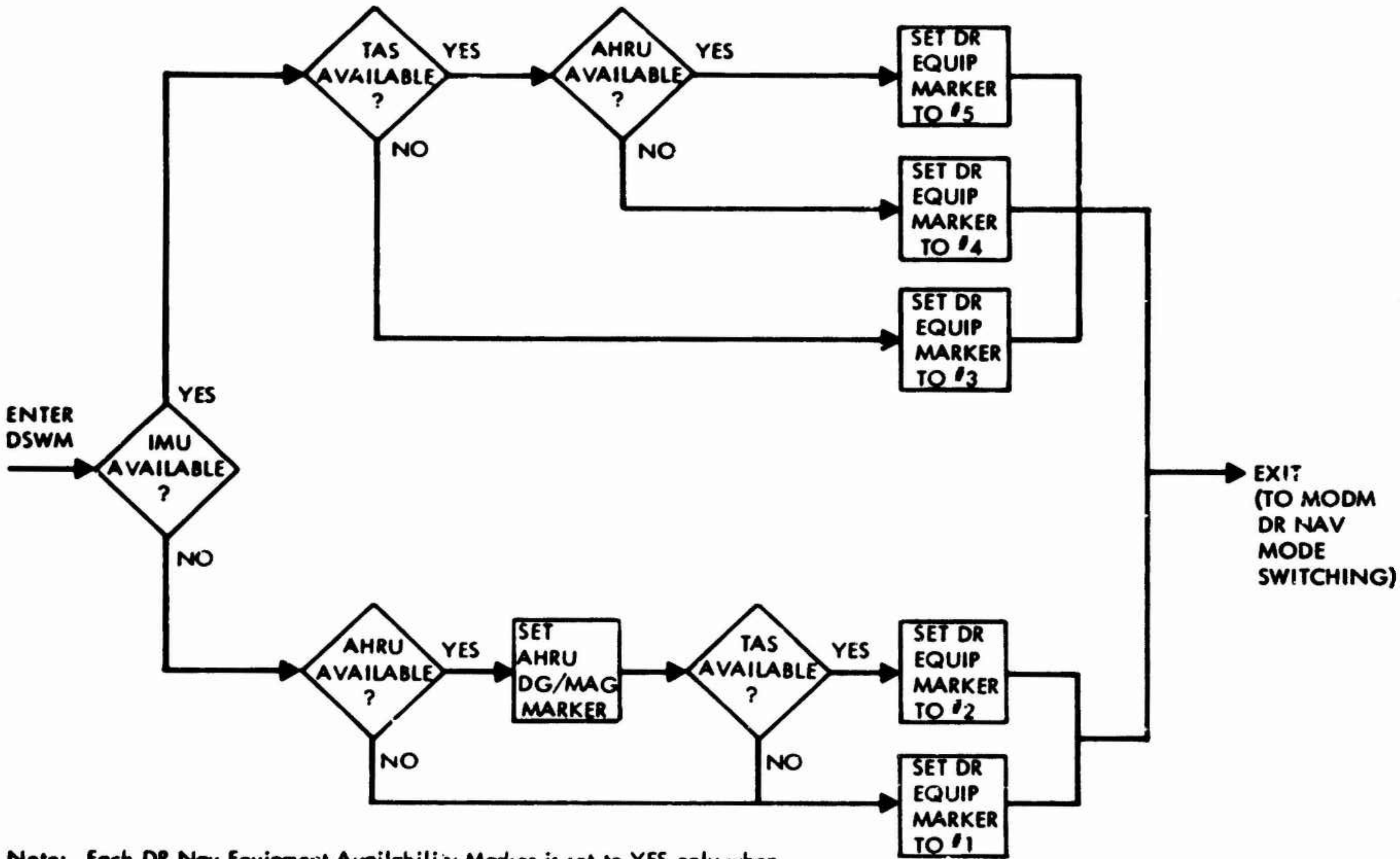


DEQM = DR Nav Equipment Configuration Marker

DRMM = DR Mode Marker

40884

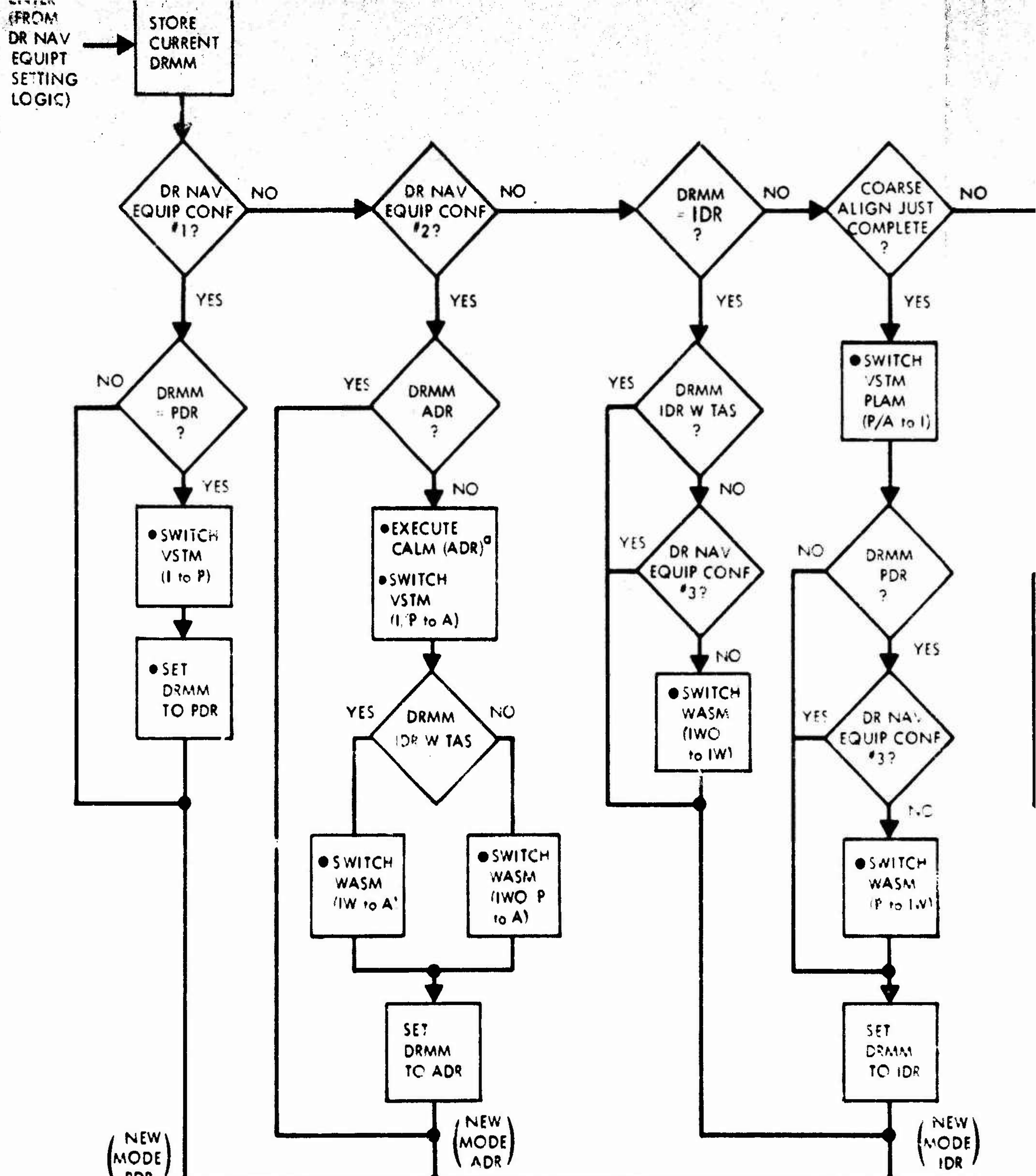
Figure 25. Overall DSWM Organization/Logic/Data Flow



Note: Each DR Nav Equipment Availability Marker is set to YES only when (a) that equipment is producing valid data, and (b) its use has been commanded by the operator by his appropriate control action.

40888

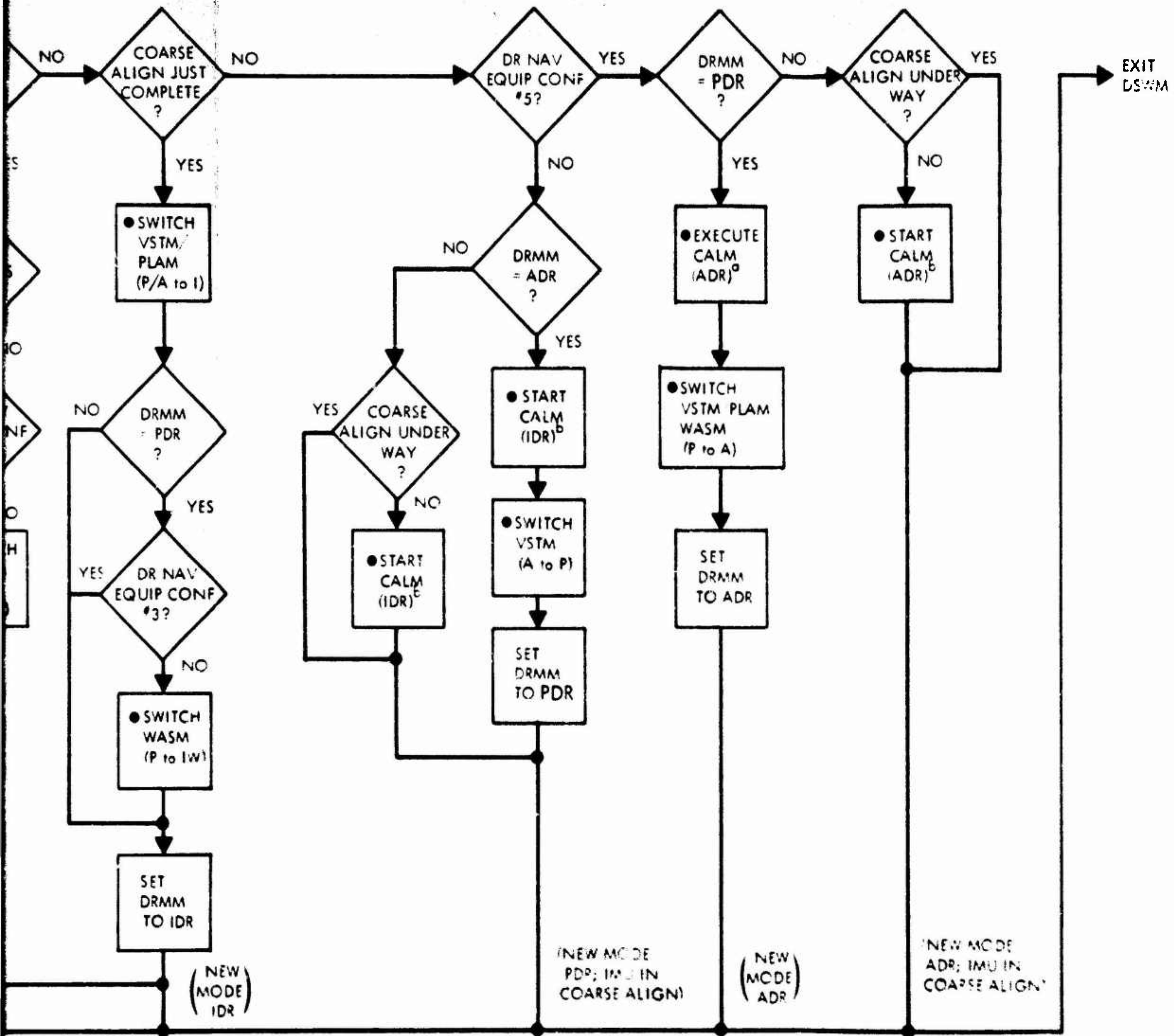
Figure 26. DR Equipment Configuration Marker Setting



(a) I.e., set CALM ADR Coarse Align Complete Marker to NO.
(b) I.e., set CALM IDR Coarse Align Complete Marker to NO.

FIGURE 37 DR NAV MODE SWITCHING REQUIREMENTS SUMMARY

2



Note: DR Nav. Module switching requirements are summarized in Table [illegible].

40893

UMMARY

TABLE LXIII. DR NAV MODE OVERALL SWITCHING REQUIREMENTS SUMMARY (DR NAV MODULES SWITCHING)

<table>
<tr><th rowspan="3">Last DR Mode/ Align Status
Current DR Equipment Configuration</th><th colspan="2">IDR</th><th colspan="3">ADR</th><th colspan="3">PDR</th></tr>
<tr><th>With TAS</th><th>Without TAS</th><th colspan="3">CALM (IDR):</th><th colspan="3">CALM (IDR):</th></tr>
<tr><th>(Wind Comp.)</th><th>(No Wind Comp.)</th><th>Not Started</th><th>Underway</th><th>Just Complete</th><th>Not Started</th><th>Underway</th><th>Just Complete</th></tr>
<tr><td>⑤
IMU and AHRU/TAS
(Max. Poss. DR Mode = IDR with Wind Comp.)</td><td>-----------</td><td rowspan="3">• WASM Start-up (IDR w/TAS)</td><td>• Start CALM(IDR)</td><td>------</td><td rowspan="3">• DRMM=IDR
• Init.VSTM/PLAM[b]/WASM (ADR to IDR w/TAS)</td><td colspan="2">• Execute Calm(ADR)[a]
• DRMM=ADR
• Init. WASM (PDR to ADR)
• Init. VSTM/PLAM[b] (PDR to ADR)</td><td rowspan="3">• DRMM=IDR
• Init. VSTM/PLAM[b]/WASM (PDR to IDR w/TAS)</td></tr>
<tr><td rowspan="2">④
IMU/TAS Only
(Max. Poss. DR Mode=IDR with Wind Comp.)</td><td rowspan="2">------------</td><td>• Start CALM(IDR)</td><td>------</td><td rowspan="3">• Start CALM(IDR</td><td rowspan="2">------</td></tr>
<tr><td colspan="2">• DRMM=PDR
• Init. VSTM (ADR to PDR)</td></tr>
<tr><td>③
IMU Only or IMU/AHRU Only
(Max. Poss. DR Mode=IDR without Wind Comp.)</td><td>------------</td><td>-----------</td><td colspan="2">Same as for Equipment Configuration No. 4 Above</td><td>• DRMM=IDR
• Init. VSTM/PLAM[b](ADR to IDR w/o TAS)</td><td>------</td><td>• DRMM=IDR
• Init. VSTM/PLAM[b](PDR to IDR w/o TAS)</td></tr>
<tr><td rowspan="2">②
AHRU/TAS Only
(Max. Poss. DR Mode=ADR)</td><td colspan="2">• Exec. CALM(ADR)[a] • DRMM=ADR
• Init. VSTM/PLAM[b] (IDR to ADR)</td><td colspan="3" rowspan="2">--------</td><td colspan="3" rowspan="2">• Execute CALM(ADR)[a]
• DRMM=ADR
• Init. WASM (PDR to ADR)
• Init. VSTM/PLAM[b](PDR to ADR)</td></tr>
<tr><td>• Init. WASM (IDR with TAS to ADR)</td><td>• Init. WASM (IDR w/o TAS to ADR)</td></tr>
<tr><td>①
AHRU Only or TAS Only or No Equipment</td><td colspan="2">• DRMM=PDR
• Init. VSTM (IDR to PDR)</td><td colspan="3">• DRMM=PDR
• Init. VSTM (ADR to PDR)</td><td colspan="3">--------</td></tr>
</table>

(a) This table assumes all CALM(ADR) equations are executed in a single fast loop (MCDM) cycle, just following DSWM execution on that cycle.

DRMM = DR Nav Mode Mark
---- = No Action Requi:

(b) Except PLAM variables $T_{P/C}, T_{P/L}, T_{L/C}$ (initialized by CALM)

40996

TABLE LXIV. DR NAV MODULES VARIABLES/EXECUTION ORDER SWITCHING REQUIREMENTS

To \ From		IDR (F or S)		ADR (DG or Mag)	PDR
		with TAS	without TAS		
IDR (F or S)	With TAS	------------	WASM: $\lvert\Delta v_{ASM}\rvert K = 0$	VSTM: f=-g, BYPASS (I1) PLAM: $f_P=0, (\omega_{P/I})_P=0,$ $\Delta f=0, \Delta\omega=0,$ $\Delta f_K=0, \Delta\omega_K=0,$ $\omega_K=0$ (IF Only)	WASM: $\lvert\Delta v_{ASM}\rvert K = 0$ VSTM: Same as ADR to IDR PLAM: " " "
	Without TAS	------------	------------		VSTM: Same as ADR to IDR PLAM: " " "
ADR (DG or MAG)		WASM: Execute (I1,I2) instead of (A1,A2) VSTM: BYPASS (A1) PLAM: $\omega_K=0$	WASM: $\lvert\Delta v_{ASM}\rvert K=0$, Execute (I1,I2) instead of (A1,A2) VSTM: BYPASS (A1)	------------	WASM: Same as IDR (w/o TAS) to ADR VSTM: " " PLAM: $\omega_K=0$
PDR		VSTM: $\beta_L=0$		VSTM: Same as IDR to PDR	----------

Note: After the above initializations, execution of each DR module should begin with the first new-mode operation and continue according to the order shown in the module operations summary table (except for the first-cycle omissions and substitutions above). The DR modules must also be executed in the order VSTM, PLAM, WASM.

III.5.e

REFERENCE NAVIGATION MEASUREMENT SWITCHING MODULE

(RSWM)

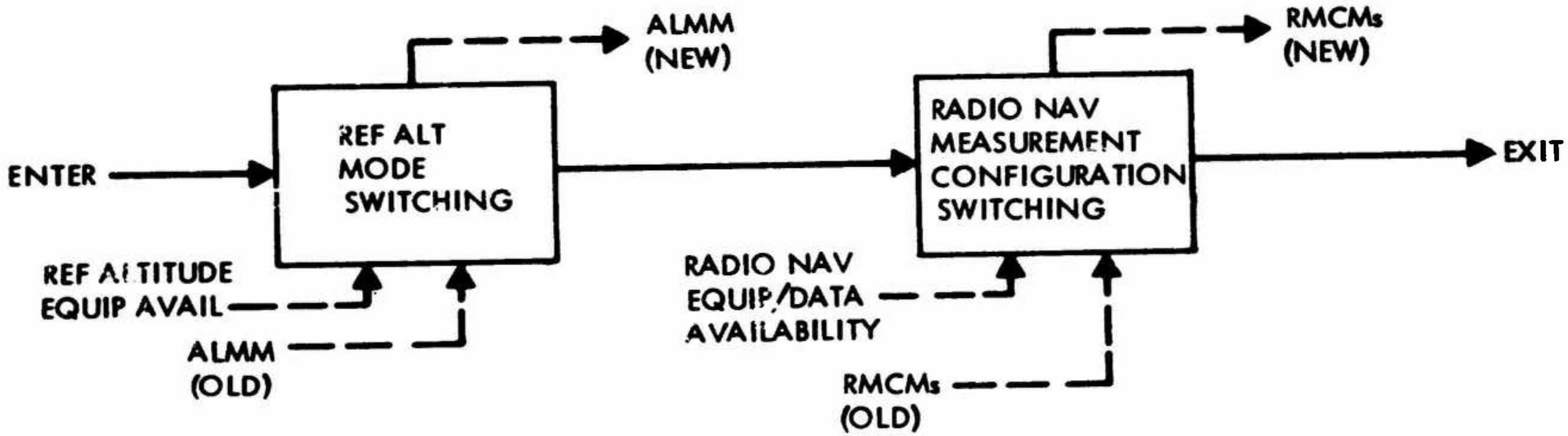
SPECIFICATION

This module comprises the operations necessary to switch processor operation from one reference navigation-measurement availability/use configuration to another such configuration, in the event of new measurement availability or old measurement drop-out.

Separate submodules are included for switching of (a) the TDPM emitter range/range-rate measurement configuration, and (b) the reference altitude measurement (ALTM) mode of operation.

In particular, the TDPM switching is organized first on a general, overall emitter-net basis, and then into a more detailed per-emitter basis. Markers are generated for each emitter which control the switching and subsequent operation of not only the TDPM modules themselves, but their associated KF error substates as well (via the KSWM).

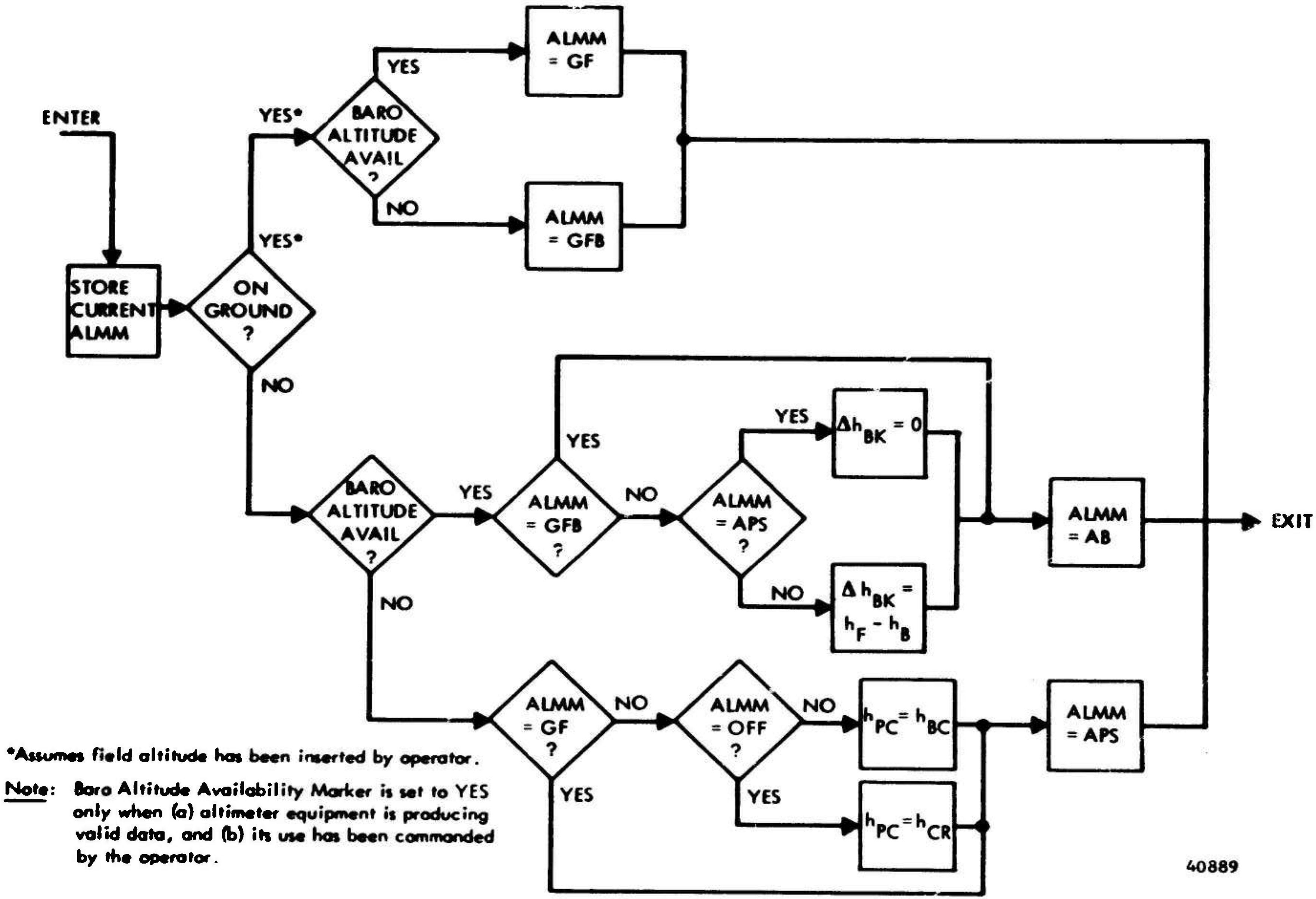
The ALTM switching sets a single marker to corresponding control switching and subsequent operation of the ALTM and its attendant KF error substate.



ALMM = Ref Altitude Mode Marker

RMCM = Radio Measurement Configuration Markers

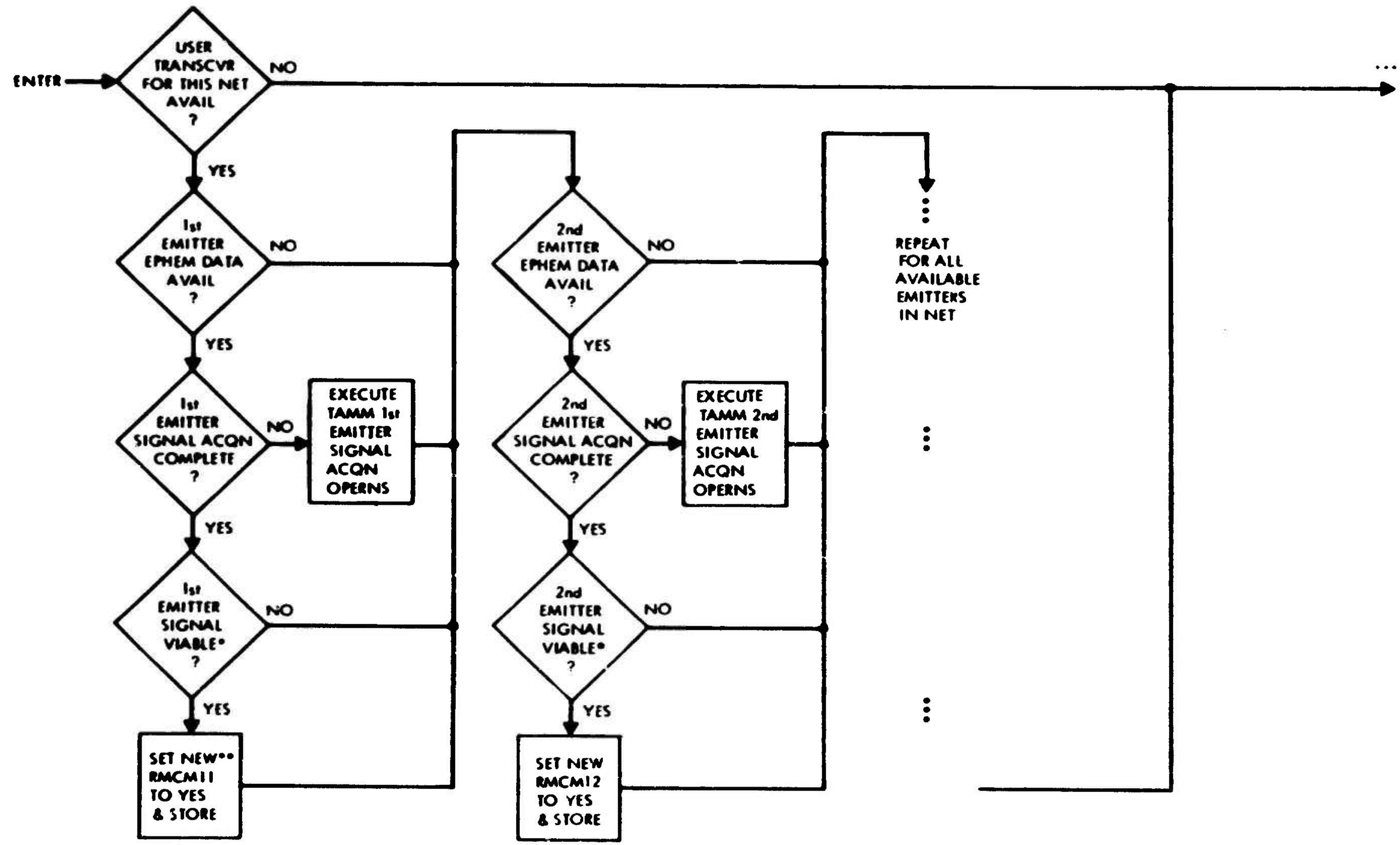
Figure 28. Overall RSWM Organization/Logic/Data Flow



*Assumes field altitude has been inserted by operator.

Note: Baro Altitude Availability Marker is set to YES only when (a) altimeter equipment is producing valid data, and (b) its use has been commanded by the operator.

Figure 29. Reference Altitude Mode Switching Logic



* I.e., are signal-to-noise ratio and LOS elevation angle sufficient ?
** RMCMnj = range/range rate measurement availability marker for jth emitter of nth net.
*** Repeat this logic for each remaining net and then exit routine.

Figure 30. TDPM Measurement Availability/Marker Setting

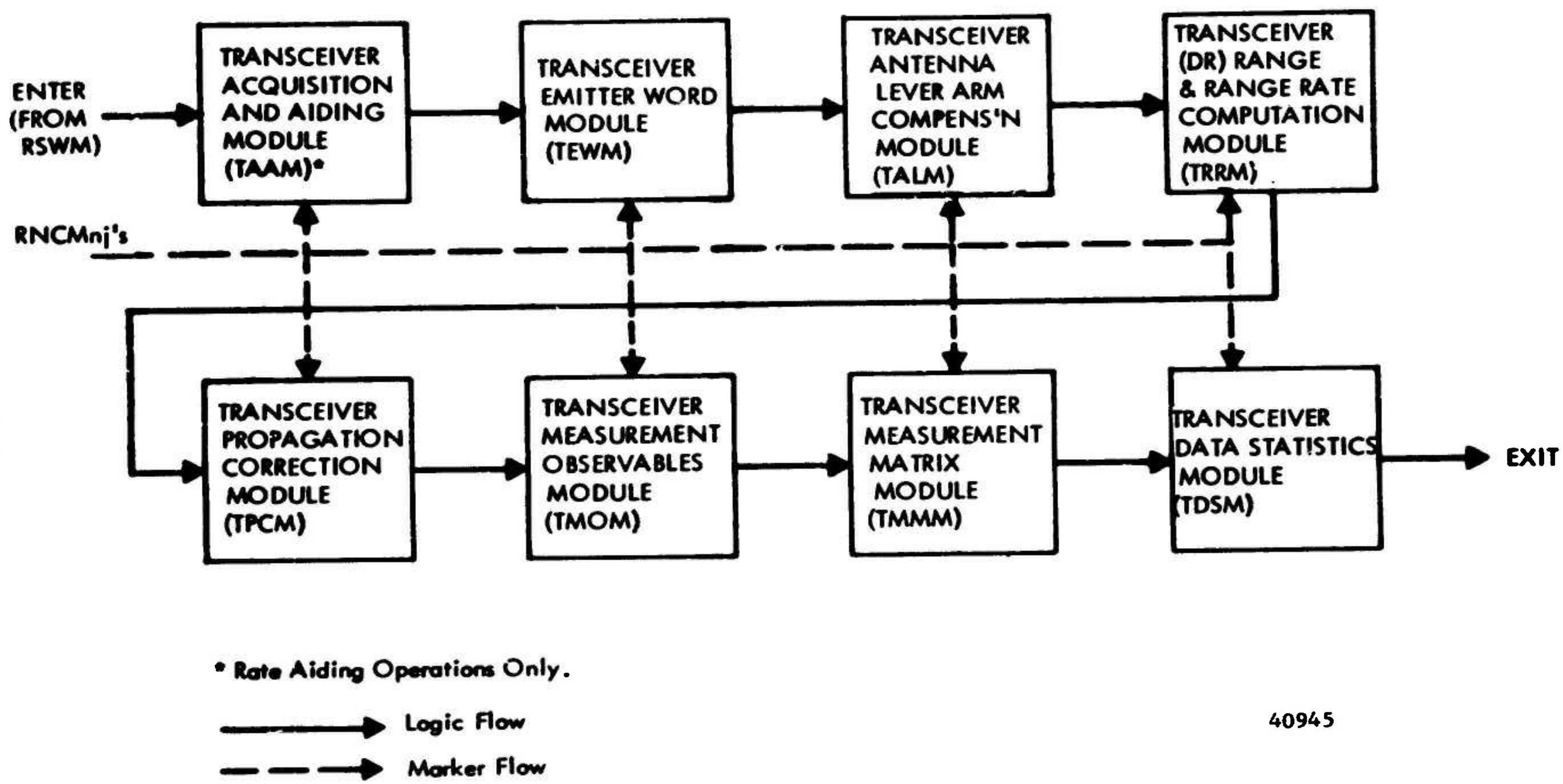


Figure 31. TDPM Modules Processing Flow

III.5.f

KALMAN FILTER CONFIGURATION SWITCHING* MODULE

(KSWM)

SPECIFICATION

*Navigation configuration and C Frame switching

This module, using the DR navigation mode and reference navigation configuration markers as inputs, controls the operation and switching of KF substates so as to correspond to and synchronize with the operation and switching of the D and R modules.

To facilitate this, modules are grouped into (a) those dealing with current-cycle data (KTMM, KMMM), (b) those dealing with last-cycle data (KTUM, KFIM, KMRM, KMCM, and KMOM), and (c) those dealing with data from both cycles (KCOM). For each of these modules, both a standard operation (i.e., operation according to the MLI module specification), and a nonstandard operation (i.e., slightly modified operation, or complete bypass) is defined. Standard or nonstandard current-cycle and next-cycle operation, and appropriate KF substate switching, are then prescribed as a function of all possible combinations of D mode, R configuration, or C frame changes (or none at all) in the last and current KF cycles.

In particular, the substate switching requirements are simplified by defining them in terms of requirements on three D-substate, and three R-substate, logical sets; i.e., all those substates common to the pre- and post-switch (D or R) configurations, and all those associated with only the old, or only the new, configurations.

The rationale underlying KF module operation (standard, nonstandard, or bypass) during processor configuration switching rests simply on the desire to maintain the KF estimate current through such periods. Since switching operations take time, and estimator currency resides in the KTMM and KTUM module operations, operations of all other modules are reduced or bypassed to allow sufficient time for intermodal switching, and intramodal execution, of these modules at a sufficiently high rate.

TABLE LXV. KSWM nth CYCLE OVERALL REQUIREMENT SUMMARY

(A) KF MODULE TEMPORAL GROUPING SUMMARY

Module Temporal Group	Module Name	Module Mnemonic
Current-Cycle Data Processing Modules	Time Update Matrix Generation Measurement Matrix Generation	KTMM KMMM
Last-Cycle Data Processing Modules	Est/Cov Matrix Time Update * Prefiltering { Reasonableness, Linear Combination, Optimal Selection } Est/Cov Matrix Filtering	KTUM KMRM KMCM KMSM KFIM
Mixed Cycle DPM	E/ /Processor Control	KCON

*KFIM Modules

(B) nth CYCLE KF MODULE OPERATIONS MODIFICATIONS FOR NAV CONFIGURATION CHANGES

KTMM & KTAM Modules	If Current (nth) KF Cycle Involves:	
	No D or R Conf. Change	Change in D or R or Both
KTMM	STD (Dn,Rn)	NSTD (D'n,R'n)
KTAM	STD (Dn,Rn,Mn)	NSTD (D'n,R'n,M'n)[b]

KTUM & KFIMs Modules	If Last (n-1st) KF Cycle Involved:	
	No D or R Conf. Change	Change in D or R or Both
KTUM	STD(Dn-1,Rn-1)	NSTD(D'n-1, R'n-1)
KFIMs	STD(Dn-1,Rn-1,Mn-1)	NSTD(D'n-1,R'n-1,M'n-1)[b]

KCON Module		If Current (nth Cycle) Involves:	
		No D or R Conf. Change	Change in D or R or Both
If Last (n-1st) Cycle Involved	No Change	STD(Dn-1,Rn-1;Dn,Rn)	NSTD(Dn-1,Rn-1;D'n,R'n)[a]
	Change	STD(D'n-1,R'n-1;Dn,Rn)[a]	NSTD(D'n-1,R'n-1;D'n,R'n)[a]

(C) KF C FRAME SWITCHING REQUIREMENTS

If Last (n-1st) KF Cycle Involved:	
No C Frame Change Command	C Frame Change Command
• Bypass C Frame Switching Operations • Set up nth Cycle KF Module Operations per Table B	Set up nth Cycle KF Module Operations per Table B, except: ■ Bypass KCON ● Execute C Frame Switching Operations just after nth Cycle KFIMs execution (or just after KTUM if KFIMs are bypassed)[c]

c - Assumes that C Frame switching of non-KF variables (see CSWM specification) is delayed until completion of all operations in KF cycle in which operator initiates C frame change command.

Notes:

STD - Standard module operation (per module specification)

NSTD - Modifications of standard module operations required to accommodate DR and/or Ref Nav. configuration changes within a KF cycle

Dn,Rn,Mn - Single, nth cycle D substate, R substate, or measurement configuration required to be processed when no D, R, or M changes occur in nth cycle.

D'n,R'n,M'n - Multiple, nth cycle D substate, R substate, or measurement configurations required to be processed when D, R, or M changes occur in nth cycle.

a - Candidates for complete bypass (except set $u = 0, \dot{u} = 0$) for any configuration change.

b - Candidates for complete bypass for D (or simultaneous D and R changes) configuration change only.

As a special requirement on the very first complete KF cycle following turn-on, all KF modules should be entirely bypassed except the current-cycle data processing (i.e., the KTMM and KMMM) modules.

40997

TABLE LXVI. SUMMARY OF nth-CYCLE NAV CONFIGURATION SWITCHING MODIFICATIONS TO STANDARD KTMM OPERATIONS

Operation			D Substates	R Substates
Execute Once at each Current-Cycle Change of:	DR Mode or R Conf	Substate Switching Set Formation	• Identify, Label and Store the D Substate Sets[c]: PRED, POSTD, COMD[a]	• Identify, Label, and Store the R Substate Sets[d]: PRER, POSTR, COMR[a]
	DR Nav Mode	Preswitch Time Update Matrix Storage	• Store Preswitch-Generated: $(\phi_{Dss'}, G_{Dss'}, R_{Dss'})_{s=COMD,\ s'=PRED+COMD}$	• Store Preswitch-Generated: $(\phi_{Rs}, R_{Rs})_{s=COMR}$[b]
		Postswitch Time Update Matrix Generation	• Execute Standard Postswitch Generation of: $(\phi_{Dss'}, G_{Dss'}, R_{Dss'})_{s,s'=POST+COMD}$	• Execute Standard Postswitch Generation of: $(\phi_{Rs}, R_{Rs})_{s=COMR}$[b]
	Ref Nav Configuration	Preswitch Time Update Matrix Storage	• Store Preswitch-Generated: $(\phi_{Dss'}, G_{Dss'}, R_{Dss'})_{s,s'=COMD}$[b]	• Store Preswitch-Generated: $(\phi_{Rs}, R_{Rs})_{s=COMR}$
		Postswitch Time Update Matrix Generation	• Execute Standard Postswitch Generation of: $(\phi_{Dss'}, G_{Dss'}, R_{Dss'})_{s,s'=COMD}$[b]	• Execute Standard Postswitch Generation of: $(\phi_{Rs}, R_{Rs})_{s=COMR}$

a. PRED(R), POSTD(R), COMD(R) = Preswitch-only, postswitch-only, and common preswitch/postswitch D(R) substates.
b. For these configuration change situations, PRER=POSTR=Null.
c. This operation is based on use of preswitch and postswitch DRMMs.
d. This operation is based on use of preswitch and postswitch RMCMs and ALMMs.

TABLE LXVII. SUMMARY OF nth CYCLE NAV CONFIGURATION SWITCHING MODIFICATIONS TO STANDARD KTAM AND KFIM OPERATIONS

<table>
<tr><th colspan="5">A. KTMM MODIFICATIONS</th></tr>
<tr><th colspan="3">Operations \ Substate Class</th><th>D Substates</th><th>R Substates</th></tr>
<tr><td rowspan="4">Execute Once at Each Current Cycle Change of:</td><td>DR Nav Mode</td><td>Standard Operations Bypass</td><td colspan="2">• Completely bypass all further current cycle KTMM operations.</td></tr>
<tr><td rowspan="3">Ref Nav Conf.</td><td rowspan="3">TSMM Set* Generation Modifications</td><td colspan="2">• Discontinue further generation of all TSMM sets involving PRER substates.</td></tr>
<tr><td colspan="2">• Initiate postswitch generation of all TSMM sets involving POSTR substates.</td></tr>
<tr><td colspan="2">• (Continue uninterrupted generation of all TSMM sets involving only COMR substates)</td></tr>
<tr><th colspan="5">B. KFIMs MODIFICATIONS</th></tr>
<tr><th colspan="3">Operation \ Substate Class</th><th>D Substates</th><th>R Substates</th></tr>
<tr><td rowspan="2">Execute Once in Current KF Cycle if in last KF Cycle:</td><td>DR Nav Mode Change</td><td>Standard Operations Bypass</td><td colspan="2">• Completely bypass all current cycle KFIM operations</td></tr>
<tr><td>R Nav Conf. Change</td><td>TSMM Set Use Modifi-cations</td><td colspan="2">• Omit processing of all TSMM sets* whose last-cycle KMMM generation was discontinued.</td></tr>
</table>

*Time Synchronized Measurement Matrix Set: $(\bar{Y}_m, \overline{\Delta Y}_m, \bar{C}_m, \bar{M}_{ms}, \bar{N}_{ms}, \bar{Z}_{ms}, \bar{W}_{ms})$

TABLE LXVIII. SUMMARY OF nth CYCLE NAV CONFIGURATION SWITCHING MODIFICATIONS TO STANDARD KTUM OPERATIONS

Operation \ Substate Class		D Substates	R Substates	D/R Substates
Execute Once for Each Nav Configuration Change (D or R) in Last KF Cycle	Time Update Segment Initialization	<u>s,s' = COMD:</u> $(x_{Ds}, P_{Dss'})_{\text{Post Switch}} = (x_{Ds}, P_{Dss'})_{\text{Pre Switch}}$ <u>s,s' = POSTD:</u> $(x_{Ds}, P_{Dss'}, u_{Ds})_{\text{Post Switch}} =$ Stored Constants	<u>s,s' = COMR:</u> $(x_{Rs}, P_{Rss'})_{\text{Post Switch}} = (x_{Rs}, P_{Rss'})_{\text{Pre Switch}}$ <u>s,s' = POSTR:</u> $(x_{Rs}, P_{Rss'})_{\text{Post Switch}} =$ Stored Constants	<u>s = COMD, s' = COMR:</u> $(P_{D/Rss'})_{\text{Post Switch}} = (P_{D/Rss'})_{\text{Pre Switch}}$ <u>s = POSTD, s' = POSTR:</u> $(P_{D/Rss'})_{\text{Post Switch}} =$ Stored Constants
	Time Update Segment Execution	<u>s,s' = POSTD + COMD:</u> Execute Std KTUM D Operations, Using Update-Interval-Corresponding: $(\phi_{Dss'}, G_{Dss'}, R_{Dss'})$ from Last-KF-Cycle, Nonstandard KTMM D Operations	<u>s,s' = POSTR + COMR:</u> Execute Std KTUM R Operations, Using Update-Interval-Corresponding: (ϕ_{Rs}, R_{Rs}) from Last-KF-Cycle, Nonstandard KTMM R Operations	<u>s,i = POSTD + COMD, s' = POSTR + COMR</u> Execute Std KTUM D/R Operations, Using Update-Interval-Corresponding: $(\phi_{Dsi}, G_{Dsi}, R_{Dsi};$ $\phi_{Rs'}, R_{Rs'})$ from Last-KF-Cycle, Nonstandard KTMM D and R Operations

40998

TABLE LXIX. SUMMARY OF KF DR NAV SUBSTATE SWITCHING SETS

To DR Nav Mode		From DR Nav Mode: IDR	PDR	ADR(DG)	ADR(MAG)
To DR Nav Mode	IDR	Com: DI1-7	DI1,2=DP1,2	DI1,2=DA1,2	DI1,2=DA1,2
		Pre: "	DP3	DA3-7	DA3-7
		Post: "	DI3-7	DI3-7	DI3-7
	PDR	DP1,2=DI1,2	DP1-3	DP1,2-DA1,2	DP1,2-DA1,2
		DI3-7	"	DA3-7	DA3-7
		DP3	"	DP3	DP3
	ADR(DG)	DA1,2=DI1,2	DA1,2=DP1,2	DA1-7	DA1-7
		DI3-7	DP3	"	"
		DA3-7	DA3-7	"	"
	ADR(MAG)	DA1,2=DI1,2	DA1,2=DP1,2	DA1-7	DA1-7
		DI3-7	DP3	"	"
		DA3-7	DA3-7	"	"

Note: New mode substate initialization requirements:

- New common substates = old common substates
- Post substates = stored constants

Pre, Post, Com = Preswitch-only, postswitch-only, preswitch-postswitch-common substate sets

TABLE LXX. SUMMARY OF KF REF ALTITUDE SUBSTATE SWITCHING SETS

To Ref Alt Mode	From Ref Altitude Mode: APS	AB	GFB	GF
APS	COM δh_p PRE " POST "	--- δh_B δh_p *	--- --- δh_p *	--- --- δh_p
AB	----- ----- δh_B	δh_B " "	---- ---- δh_B	
GFB	---	---	----	
GF	δh_p ---	δh_B ---	---- ----	

*Initialize According to $\delta h_p = \delta h_B$

**Assumes candidate KF ref alt error substate composition:

$$x_{RUA} = \begin{bmatrix} \delta h_B \\ \delta h_p \end{bmatrix} = \begin{cases} \text{Barometric Altitude Error} \\ \text{Pseudo-Altitude Error} \end{cases}$$

Note: New mode substate initialization requirements:

- New common substates = old common substates
- Post substates = stored constants

Pre, Post, Com = Preswitch-only, postswitch-only, preswitch-postswitch-common substate sets

TABLE LXXI. KSWM KF R SUBSTATE SWITCHING SETS*

Current RMCMnj Marker Setting \ Old RMCMnj Marker Setting		njth Emitter Measurement Was: Not Available	njth Emitter Measurement Was: Available
njth Emitter Measurement is:	Not Available	---------------	njth signal KF error substate is a preswitch-only R substate
	Available	njth signal KF error substate is a postswitch-only R substate	njth signal KF error substate is a pre- and postswitch-common R substate**

*Applies to reference altitude measurements as well as njth emitter measurements.
**If KF modelling of njth signal error is to be omitted (to make room for a new signal substate), this logical output marker state should be artificially set to preswitch-only instead, to enable its subsequent omission.

TABLE LXXII. C FRAME SWITCHING OPERATIONS SUMMARY (KF MODULES)

Operation \ Substate Class		D Substates			R Substates	D/R Substates		
		IDR	PDR	ADR	s,s' = All REnE** in sr0	IDR	PDR	ADR
Execution in nth KF Cycle*	Kalman Filter Vectors and Matrices Switching	s=DI1+DI2	s=DP1+DP2	s-DA1+DA2		s=DI1+DI2	s=DP1+DP2	s=DA1+DA2
		$(x_{Ds})_n = T'_{SW}(x_{Ds})_n$ $(u_{Ds})_{n-1} = T'_{SW}(u_{Ds})_{n-1}$ $(P_{Dss})_n = T'_{SW}(P_{Dss})_n T'^{T}_{SW}$			$(x_{Rs})_n = T'_{SW}(x_{Rs})_n$ $(u_{Rs})_{n-1} = T'_{SW}(u_{Rs})_{n-1}$ $(P_{Rss'})_n = T'_{SW}(P_{Rss'})_n T'^{T}_{SW}$	s' = All sr0 except REnE Subset $(P_{D/Rss'})_n = T'_{SW}(P_{D/Rss'})_n$ $(P_{D/Rs's})_n = (P^{T}_{D/Rss'})_n$		
		s'=DI3-DI7	s'=DP3	s'=DA3-DA7	s = All REnE in sr0 s'= All other sr0	s'=DI3-DI7	s'=DP3	s'=DA3=DA7
		$(P_{Dss'})_n = T'_{SW}(P_{Dss'})_n$ $(P_{Ds's})_n = (P^{T}_{Dss'})_n$			$\left.\begin{array}{l}(P_{Rss'})_n = T'_{SW}(P_{Rss'})_n\\(P_{Rs's})_n = (P_{Rss'})^{T}_n\end{array}\right\}(s'>s)$	s = All REnE in sr0 $(P_{D/Rss'})_n = (P_{D/Rss'})_n T'^{T}_{SW}$ $(P_{D/Rs's})_n = (P^{T}_{D/Rss'})_n$		
						s=DI1+DI2	s=DP1+DP2	s=DA1+DA2
						s = All REnE in sr0 $(P_{D/Rss'})_n = T'_{SW}(P_{D/Rss'})_n T'^{T}_{SW}$ $(P_{D/Rs's})_n = (P_{D/Rss'})_n$		

*Just after completion of nth cycle Estimate/Covariance Matrix Filtering Operations (unless bypassed on nth cycle; then just after completion of nth cycle Estimate/Covariance Matrix Time Update Operations). New C Frame commanded by operator in (n-1)st KF cycle, but non-KF Nav Variables Switching delayed until all (n-1)st cycle KF operations completed.

**All REnE assumed here to be composed of both position and velocity error substates.

Note: $T'_{SW} = \begin{vmatrix} T_{SW} & 0 \\ 0 & T_{SW} \end{vmatrix}$

III.5.g

NAVIGATION CONSTANTS INITIALIZATION MODULE

(CONM)

SPECIFICATION

This module reads in all constants necessary to overall processor modular operations. Many of these constants are navigation equipment-related (they correspond to and are needed for characterization and correction of the navigation sensor equipment complement used), while the remainder are required whatever the equipment complement employed. These requirements are broadly summarized for each processor module in the tables included in this specification.

No attempt is made here to define either the associated input data constant verification operations or their attendant control/display panel interfacing operations, since these are outside the scope of the processor developed to date.

TABLE LXXIII. D MODULE CONSTANTS REQUIREMENTS

Constants Type / D Module	DR Nav Equipment-Related Constants			Equipment Independent Constants
	IMU	AHRU	CADS(TAS)	
VSTM	----------	----------	----------	• Gravity and sub-aircraft position vector formulae constants • Pseudoacceleration model constants
PLAM	IMU inertial instrument and attitude readout calibration constants	AHRU attitude readout calibration constants	----------	----------
	Geoidal curvature formula constants			
WASM	----------	----------	TAS instrument calibration constants	• Angle-of-attack compensation function constants • Wind model constants

TABLE LXXIV. R MODULE CONSTANTS REQUIREMENTS

Constants Type / R Module	Ref Nav Equipment-Related Constants		Equipment Independent Constants
	CADS (Baro Alt)	Transceivers/Emitters	
ALTM	Barometric altimeter calibration constants	----------------	Pseudo altitude operation constants
POSM	---------	----------------	External-to-internal coordinate conversion constants
TDPM	---------	Emitter, transceiver power, clock, delay, noise, and multipath constants and statistics	Ionospheric and tropospheric constants and statistics, and phase-to-range and receiver-to-computer units conversion constants

TABLE LXXV. K MODULE CONSTANTS REQUIREMENTS

Constants Type / K Module	DR Nav Equipment-Related Constants			Equipment Independent Constants	Ref Nav Equipment-Related Constants		Equipment Independent Constants
	IMU	AHRU	CADS(TAS)		CADS(Baro Alt)	Transceivers/Emitters	
KF Structure	• IMU subst ident, dim, dyn, noise, constants	• AHRU subst ident, dim, dyn, noise, constants	• CADS(TAS) subst ident, dim, dyn, noise constants	• Pos/vel'y subst dim, dyn, noise constsnts • D substate measmt mtx constants • Pseudoacc'n error model constants	• CADS(B. Alt) subst ident, dim, dyn, noiss constants • CADS subst measmt mtx constants	• Transcvr/emitter substate ident, dimensions, dynamical, noise constants • Transcvr/emitter subst measmt mtx constants	----------
KF Timing	• KF cycle time • Module sequencing timing constants						
KTUM	*						
KFIM	• Kalman/least squares gain mix constants*						
KCOM	• IMU torquing algorithm constant	*					
KMRM	• KF reasonableness test gain constants*						
KMOM	• FOM/state relationship matrix constants*						
KMCM	• Measurement-combination relative weighting function constants • Measurement-combination noise function constants*						
KTMM	*						
KMMM	*						

*Plus constants specified under KF structure and/or KF timing above.

40999

TABLE LXXVI. INITIALIZATION/SWITCHING MODULE CONSTANTS REQUIREMENTS

Constants Type / Init/Switching Module	DR Nav Equipment-Related Constants			Equipment Independent Constants	Ref Nav Equipment-Related Constants		Equipment Independent Constants
	IMU	AHRU	CADS(TAS)		CADS (Baro Alt)	Transceivers/Emitter	
NSTM	----------------------------------			• VSTM PDR and KFM/PDR init constants • Earth rate	--		
CSWM	• New C frame definition function constants						
CALM	• VST accuracy, CALM entry test constants		--				
	• Platform leveling op'n const • Al't compl'n test constants	-----------					
DSWM	• New-IDR-mode IMU variable initialization constants	• New-ADR-mode AHRU variable initializ'n constants	• New-ADR-or IDR/TAS mode var initializ'n constants	• New-PDR-mode var init const • New-ADR-mode wind var init constants	--		
RSWM	--				• New-Baro-Alt-mode var init constants	• New-emitter configuration variable initialization constants	----------
KSWM	• New-IDR-mode IMU subst initialization constants	• New-ADR-mode AHRU subst init constants	• New-ADR-or IDR/TAS-mode TAS subst init constants	• New-PDR-mode psdo acc/n subst init const • New ADR-mode wind subst init constants	• New-Baro-alt-mode Baro alt subst init constants	• New-emitter configuration substate initialization constants	----------

41000

SECTION IV

HOL PROCESSOR (LIMITED MLI PROCESSOR VERSION)

This final main section presents and discusses a specialized, functionally limited, FORTRAN IV/IBM 370 programmed version of the MLI processor, as far as it has been developed to date.

The purposes of this development were two-fold. Both have been fulfilled. These were (a) to provide a first-time vehicle for MLI-based programming of a specific processor application in a specific language for a specific machine, and (b) to provide a program nucleus which could eventually be easily developed into either a processor algorithm evaluation, a simulation program, or a master, HOL navigation processor software generation program.

In order to permit a meaningful level of development of an HOL (FORTRAN IV) processor program toward these ends within the time and funding available in Phase II, a set of simplifying and facilitating guidelines and assumptions was therefore formulated prior to initiation of the actual programming effort. These guidelines and assumptions were then carefully adhered to during the development itself.

In this connection, a loose scenario for a multiphase tactical mission was first formulated. This scenario was chosen especially to enable (or even require) exercising many of the MLI processor functional capabilities, including most of those of special interest and importance (e.g., C-frame switching, simultaneous, dual-LOS-net pseudorange processing, etc.). Next, a candidate navigation processing design was formulated -- with particular attention to minimizing the required Kalman filter state vector size -- to accomplish this mission. This design, together with the mission scenario to which it applies, is summarized in paragraph 1.a in terms of the navigation/environment equipment assumptions, and in Table LXXVII in terms of candidate processor configuration by mission phase.

This scenario/candidate processing design then furnished the basis for identifying the appropriate subset of the general (Section III) MLI specifications necessary to implement it. To further delimit the required programming effort to the time and funding available, a set of additional simplification constraints, which are summarized in paragraph 1.b, was invoked. These were selected specifically to have minimal effect on program capabilities with respect to the chosen scenario (e.g., omission of coarse alignment and DR navigation mode switching capability did not reduce the capability of the program to conduct all significant phases of the mission shown in Table LXXVII. The residual MLI specifications required for final programming are summarized in subsection 2.

TABLE LXXVII. CANDIDATE NAV PROCESSOR CONFIGURATION BY MISSION PHASE

Processor Configuration / Mission Phase	C Frame	DR NAV Type	Measurement Use				KF Configuration								
					LOS Quad		D Substates				R Substates			KF State Size	Comments
			Baro Alt Use*	NAV-SAT Quad Use	Use	KF Alg.	DI1(3)	DI2(3)	DI3(3)	DI4(3)	RUA(1)	RUC(2)	RE(3 or 4)		
Phase 1 Enroute	C = E (Global)	IDR	X	X			X	X	X	Non-g Sens. (long corr'n time)	X	User/ NAV SAT Diff	NAVSAT Prop Error	19	User/NAV-SAT Clock Diff. and IMU Non-g Sens. Drift Rates Calibrated
Phase 2a Target Area Approach	C = KF (Target-Local)	IDR	X	X	X	Non-lin.	X	X	X	Non-g Sens. (short corr'n time)	X	User/ LOS Net Diff.	LOS Net Datum Error	18	User/LOS Net Clock Diff. and LOS Datum Error Calibrated
Phase 2b Target Area Weapon Delivery	C = KF	IDR			X	Lin.	X	X	X	Omit (Replace with white noise forcing DI3)			LOS Net Prop Error (or omit)	13 (or 9)	High Accuracy Weapon Delivery With Calibrated Clock Diff.
Phase 3 Return to Base**	C = E	IDR	X	X			X	X	X	Non-g Sens. (long corr'n time)	X	User/ NAVSAT Diff.	NAVSAT Prop Error	13	Same as Enroute Phase

*Via KF using linear algorithm

**Same processor configuration as enroute

41001

Use of this framework of mission and hardware-related ground rules and constraints (as summarized in Table LXXVII and paragraphs 1.a and 1.b), in part created for the analyst and programmer involved an atmosphere not unlike that surrounding an actual software development for a specific, real-time navigation system application. This was valuable as a test of the usefulness of the MLI specifications in situations of this sort, for which it was in large part designed.

The actual HOL program developed to date is presented in subsection 3, in terms of general organization, module descriptions, common and special subroutines, an input/output and level-of-checkout discussion, and finally the actual FORTRAN IV program listing itself.

1. SPECIALIZING ASSUMPTIONS AND CONSTRAINTS

This subsection (and Table LXXVII) summarize the assumptions and constraints which simplified and facilitated the development of the nucleus program.

a. Navigation Environment/Equipment Assumptions

(1) Navigation Environment

(a) Tactical aircraft

(b) Tactical multiphase mission

- Enroute (NAVSAT/IMU)
- Local area approach and weapon delivery (LOS ground net/IMU)
- Return to base (NAVSAT/IMU)

(2) Navigation Equipment

(a) Onboard aircraft

- Digital navigation computer(s)
- IMU (rotationally isolated)
- LOS ground net receiver equipment (single channel, high-accuracy clock)
- NAVSAT receiver equipment (single channel, high-accuracy clock)
- Barometric altimeter

(b) External to aircraft

- LOS ground emitter net (4 emitters; serial, synchronized transmissions; accurate, intra-net emitter location survey)
- NAVSAT net (4 emitters; serial, synchronized transmissions; accurate E frame-referenced ephemeris data transmissions)

b. <u>Processor Development Level Simplification Constraints</u>

(1) Basic Navigation Only

(a) Position/velocity vector outputs only (i.e., no aircraft attitude, attitude rate outputs)

(b) No coarse alignment

(c) No DR navigation mode switching (radio-inertial only)

(d) No intramission reference navigation switching

(2) Simplest Algorithm Choice (where intramodular choices exist)

(3) Limited Checkout (in absence of equipment/environment simulation)

(4) Geodetic Assumptions: Spherical, homogeneous earth

2. MLI-BASED SPECIFICATION

With the set of simplifying ground rules just defined, the relevant subset of the overall MLI specifications of Section II which is required as a basis for programming can, because of the modular organization of the processor, be quickly identified.

This required MLI specification subset is briefly summarized in the following paragraphs.

a. <u>Processor Module Requirements</u>

(1) Required Modules

(a) Initialization/Switching: CONM, CSWM

(b) DR Navigation: PLAM, VSTM

(c) Reference Navigation: ALTM, TDPM (TEWM, TRRM, TPCM, TMCM, TMMM)

(d) Kalman: KTUM, KFIM, KCOM, KTMM, KMMM

(2) Modules Not Required

(a) Initialization/Switching: NSTM, CALM, DSWM, RSWM, KSWM

(b) DR Navigation: WASM

(c) Reference Navigation: POSM, TDPM (TAAM, TALM, TDSM)

(d) Kalman: KMRM, KMCM, KMOM

b. Module Algorithm Requirements

(1) CONM: Constants appropriate to (F)IMU, CADS(baro altitude) ground LOS and NAVSAT signal equipment and use only, and for only those modules specified in paragraph 2.a(1) above.

(2) CSWM*

(a) IDR (without TAS) algorithms only

(b) C frame definition and input data**

(3) VSTM*

(a) IDR algorithms only

(b) g_E, P_S formulae**

(4) PLAM*

(a) IF algorithms only

(b) Omit Δw, Δf, $\Delta\omega_k$, Δf_k and ω_k computations entirely

(c) Omit $T_{A/P}$, $(\omega_{A/P})_P$ computations entirely

(d) K_L formula

(5) ALTM*

(6) TDPM*: G and S configuration algorithms only, pseudorange(no pseudorange rate) measurements only.

(a) TEWM: Operations GAS(3)and (4)only

(b) TRRM: GS(1), GAS(1) - (3) only

(c) TPCM: GAS(2) only

(d) TMCM: GAS(1), (3), (5) and (7) only

(e) TMMM: GAS(1), (2) and (5) only.

---

* See MLI module specification.

**See the special, spherical homogeneous earth-based formulae which are summarized at the end of this subsection.

(7) Kalman Filter Partitioned Structure*

(a) Error Substate Definitions:

- D Substates: IDR only; DI1 through DI4, DI4** of dimension 3

- R Substates: RUA (dimension 1)**, RUC (user/emitter net clock phase difference, dimension 2),** RE (emitter net datum error, dimension 3, or emitter signal propagation delay error, dimension 1 per emitter)**

(b) D Substate Vector/Matrix Structure:

- IDR only

- x_D through b_D: per specification

- A_D: 12,21,22,23,33,** 34,** 44**

- K_D: 4

- ϕ_D: 11,12,21,22,13,23,33,34,*44**

- G_D: 13,23,33

- R_D: 11,22,33,** 44**

- P_D: all

(c) R Substate Vector/Matrix Structure: per specification, except all submatrices of (K,A ϕ and R)**

(d) D/R Substate Vector/Matrix Structure: per specification

(e) D/R Measurement-Difference Vector/Matrix Structure:

- m = 2,3 only

- $Y, \bar{Y}, C, \bar{C}, \Delta Y, \Delta\bar{Y}$: per specification

- $\bar{Z}, \bar{W}$: all null

- M_D: DI1

- $\bar{M}_D$: m = 2: DI1,DI2; m = 3: DI1,DI2,DI3

---

* See MLI module specification.

**Possible mission-phase-dependent substates.

- $\bar{N}_D$: m = 2: all null; m = 3: DI3
- M_R: m = 2: per spec; m = 3: M_{RUC}, M_{RE}**

(8) KF Modules Timing and Sequencing Organization: per specification

(9) KTUM

(a) IDR only

(b) sd' = DI1+DI2 = DI1', DI3,** DI4**

(c) sro = RUA*,RUC*,RUE*

(10) KFIM

(a) sd',sro as above

(b) m = ml** (set of all sequential time-point measurements in KF cycle)

(11) KCOM

(a) IDR (IF only)

(b) sd',sro as above

(c) Non-KF Modules Control

- PLAM: Omit Δf_k and $\Delta\omega_k$ corrections
- WASM: Omit
- ALTM: AB algorithm only
- TDPM: User/emitter clock correction = user clock correction shown; emitter net datum error is $\Delta_K e = \Delta_K e - u_{REnE}$; emitter propagation error correction is $\Delta_K L = \Delta_K L - u_{REnj}$ (per emitter).

(12) KTMM

(a) IDR only

(b) sd', sro as above

(c) D Submatrix Generation: Execute the closed-form, non-recursive equations shown below, once per KF cycle for KTUM use, and once per measurement availability per KF cycle for KMMM use.

---

* Possible mission-phase-dependent substates

**Mission-phase-dependent

(d) R Submatrix Generation: Execute the closed-form, non-recursive equations shown below, once per KF cycle for KTUM use with $t = t_F - t_S$, and once per measurement availability per KF cycle for KMMM use with $t = t_i - t_F$.

KTUM Use

KMMM Use

$$\left\{\begin{aligned}
\phi_{D1'1'} &= I + A_{DC1'1'}\,\Delta t + \frac{1}{2}\left(A_{DC1'1'}\,\Delta t\right)^2 \\
\phi_{D1'3} &= \begin{bmatrix} \left(-\frac{1}{2}\, g\,\Delta t^2 + \Delta p - v_1\,\Delta t\right) \times T_{C/P} \\ \hline \left(g\,\Delta t + \Delta v\right) \times T_{C/P} \end{bmatrix} \\
G_{D1'3} &= \begin{bmatrix} \left(-\frac{1}{6}\, g\,\Delta t^2\right) \times T_{C/P} \\ \hline \left(-\frac{1}{2}\, g\,\Delta t^2\right) \times T_{C/P} \end{bmatrix}
\end{aligned}\right.$$

($A_{DC1'1'} = A_{DI1'1'}$; see MLI D Substate structure specifications)

$$\phi_{D33} = I - \left(\Delta\theta_{E/I} + \Delta\theta_{P/C}\right)_P \times \qquad R_{D1'1'} = \text{constant diagonal matrix}$$

$$\phi^*_{D34} = I\,\Delta t \quad (\text{or } 0) \qquad R^*_{D33} = \text{"}$$

$$\phi^*_{D44} = I \times A^*_{DC44}\,\Delta t \qquad R^*_{D44} = \text{"}$$

$$G_{D33} = I\,\Delta t \qquad A_{DC44} = \text{"}$$

$$(\Delta\theta_{E/I})_P = T_{P/C}\,\omega_{E/I}\,\Delta t$$

$$(\Delta\theta_{P/C})_P = (\omega_{P/C})_P\,\Delta t$$

	KTUM Use	KMMM Use
Δt	$t_F - t_S$	$t_i - t_F$
Δv	$v_F - v_S$	$v_i - v_F$
Δp	$p_F - p_S$	$p_i - p_F$
v_1	v_S	v_F

g, $\omega_{E/I}$, $T_{P/C}$, $\omega_{P/C}$: any values in current KF cycle

*Mission-phase-dependent

(13) KMMM

(a) $m = ml^*$ (see above)

(b) $n = 1$ (KF endpoint-synchronization only; no time smoothing)

(c) sd', sro as above

(d) Measurement Matrix Generation ($m = 2$; altitude): Execute the closed-form equations below once per measurement availability per KF cycle (i.e., once at $t = t_F$):

$$\bar{Y}_D = |h|_F \qquad \bar{Y}_R = (h_R)_F$$

$$\bar{C}_D = \text{constant} \qquad \bar{C}_R = \text{constant}$$

$$\bar{M}_{D1'} = M_{D1'F} = \left[g^T/|g| \quad 0 \right] \qquad \bar{M}_{RUA} = M_{RUA} = -1$$

$$\overline{\Delta Y} = \bar{Y}_D - \bar{Y}_R$$

(e) Measurement Matrix Generation ($m = 3$; LOS pseudorange): Execute the closed-form equations below once per measurement availability per KF cycle (i.e., once per availability of emitter j at time t_i):

$$\bar{Y}_{Dj} = |p_{ji} - e_{ji}| \qquad \bar{Y}_{Rj} = -R_{mj}$$

$$\bar{C}_D = \text{constant } j \qquad \bar{C}_{Rj} = \text{constant}$$

$$\bar{M}_{Dj1'} = M_{Dj1'i}\,\phi_{D1'1'i,F} \qquad \bar{M}^*_{RUC} = M_{RUC}\,\phi_{RUCi,F}$$

$$\bar{M}_{Dj3} = M_{Dj1'i}\,\phi_{D1'3i,F} \qquad \bar{M}^*_{REj} = M_{REji}\,\phi_{REji,F}$$

$$\bar{N}_{Dj3} = -M_{Dj1'i}G_{D1'3i,F} \qquad M^*_{RUC} = \left[-C \quad 0 \right] \text{ (C = velocity of light)}$$

$$M_{Dj1'} = \left[r_{ji}^T \quad 0 \right] \qquad M^*_{REji} = -r^T_{ji} \text{ (LOS net datum error)}$$

$$r_{ji} = \frac{p_{ji} - e_{ji}}{|p_{ji} - e_{ji}|} \qquad M_{REj} = -1 \text{ (NAVSAT or LOS emitter propagation error)}$$

$$\overline{\Delta Y}_j = \bar{Y}_{Dj} - \bar{Y}_{Rj}$$

*Mission-phase-dependent

(14) Simplified Formulae (Based on homogeneous, spherical earth assumption):

(a) Gravity

$$g_E(p_E) = -\left[\left\{\frac{k_o}{|p_E|^3} - |\omega_{E/I}|^2\right\} I + (\omega_{E/I})_E \, (\omega_{E/I})_E^T\right] p_E$$

(if $|p_E| \le R_o - \Delta R$, take $g_E = 0$

($k_o = G_o R_o^2$; G_o, R_o, ΔR are scalar constants)

(b) Subaircraft Position Vector

$$p_S = \frac{R_o}{|p_E|} \, p_E$$

(c) Earth's Geoidal Curvature Matrix

$$K_L = \frac{1}{|p_E|} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

(d) C Frame Definition

Enroute: $C = E$, so that $C_{C/E} = 0$, $T_{C/E} = I$

Objective Area: C = Tangent-plane, Cartesian frame centered at λ_{CTR}, L_{CTR}, h_{CTR} (latitude, longitude, altitude), so that:

$$C_{C/E} = (R_o + h_{CTR}) \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \end{bmatrix} \qquad T_{C/E} = \{c_{ij}\}_{3x3}$$

$c_{11} = \sin\lambda_{CTR}$ $\quad c_{12} = \cos\lambda_{CTR}\cos L_{CTR}$ $\quad c_{13} = \cos\lambda_{CTR}\sin L_{CTR}$

$c_{21} = 0$ $\quad c_{22} = -\sin\lambda_{CTR}$ $\quad c_{23} = \cos L_{CTR}$

$c_{31} = \cos\lambda_{CTR}$ $\quad c_{32} = -\sin\lambda_{CTR}\cos L_{CTR}$ $\quad c_{33} = -\sin\lambda_{CTR}\sin L_{CTR}$

3. FORTRAN IV/IBM 370 PROGRAM DESCRIPTION

The MLI specification approach to generating digital navigation system software does in fact introduce many savings in programming, in checkout, and in changes and modifications to such software. This conclusion has been reached as a result of the direct (although limited) experience gained with the technique in actually programming the limited-version, FORTRAN IV/IBM 370 program presented and discussed here. The validity of this contention will become more apparent against the background of the organizational simplicity and flexibility which, by following the MLI specification, have been embedded in this sample program.

Before proceeding with the discussion, it is first emphasized that the program developed to date is not yet either usable or fully checked out. That is, because no dynamic, mission-history simulation of navigation sensor input data and vehicle flight profile is yet available, it has not been possible to carry checkout past a relatively rudimentary level (i.e., only to the level of separate module or module group checkout and only with fixed inputs), let alone to exercise the overall program in simulated navigation. Nevertheless, the program, even in its present state, constitutes an advanced nucleus and point of departure for the eventual development of a complete processor/processor environment simulation and evaluation program.

At least equally important of course, are the facts that (a) it has, as indicated above, served as a practical test bed for the MLI specification concept and (b) its design illustrates how one might design a processor in any HOL language, either to verify (and adjust) a selected mechanization, or even to translate the selected mechanization into an actual, real-time HOL system navigation program.

a. General Organization

In programming this version, extensive use has been made of the subroutines, array structuring, and indexing features of FORTRAN IV. Dynamic forcing inputs are of course (as indicated above) not available, and output routines have not been included, since in checkout these depend on the checkout technique employed, and for any particular working application depend on the output coordinates peculiar to the application itself (e.g., latitude-longitude, UTM, etc.). On the other hand, internal, intermodular communication has been embedded (but not yet fully checked out) via storage in appropriate common blocks, organized so as to ensure that the most recently computed results will always be used.

The program has been organized into five basic modular groups: the start-up and initialization modules, the DR navigation modules, the reference modules, the Kalman modules, and the special output modules.*

*Note that these groupings differ slightly from the MLI groupings in that each of the switching modules has been embedded as part of the group of modules it controls. This is not, however, a real difference, but only one in point of view.

Of these, the D, R, and K module groups (which modularly embed the dynamic mode, configuration, and C frame switching operations) together comprise the dynamic navigation loop, and the special output modules (none of which are included) comprise the functions necessary for any of a wide class of output operations (e.g., conversion of internal C frame-referenced navigation quantities to output display coordinates or control signals).

The start-up and initialization modules, in addition to embedding the MLI CONM and NSTM functions, also configure the processor in accordance with the navigation equipment complement available (i.e., any subset of the overall IMU/AHRU/CADS/transceiver equipments which the current MLI processor is capable of processing).*

Another noteworthy (and prospectively very useful) feature of the program is its algorithm timing flexibility. Under input control, each module or algorithm thereof can optionally be executed at any desired frequency relative to a basic frequency, or even bypassed entirely.

In actual operation, all data constants and variables are saved in labeled common storage and are initialized in the block data routine. During turn-on, each basic module group is called and executed once to initialize fixed or specific arrays, pointers, algorithm timing, etc. Navigation sensor equipment characteristics, error models, and availabilities are input by the NSTM and CONM modules. These inputs are read using the FORTRAN Name List function.

After all required initial inputs are read and verified, dynamic main loop operation begins, with the DR navigation mode, the R configuration, the C frame, and the K configuration respectively and continuously controlled by the DSWM, the RSWM, the CSWM, and the KSWM. This is accomplished by setting internal pointer and software configuration markers to control the execution of only those algorithms appropriate to the selected type of processor operation.

In this functionally limited program, only the IDR mode algorithms are programmed. However, the markers and specific entry points for ADR and PDR operations have been included so that the mode-specific algorithms can later be inserted with absolutely no change to the program organization. Figure 32 is a summary block flow diagram of the program organization.

Table LXXVIII summarizes the main program organization in terms of the FORTRAN flow and command symbology actually implemented.

*It is this processor-configuration feature which, if appropriately generalized, could comprise an important step toward eventual realization of an HOL software generation program.

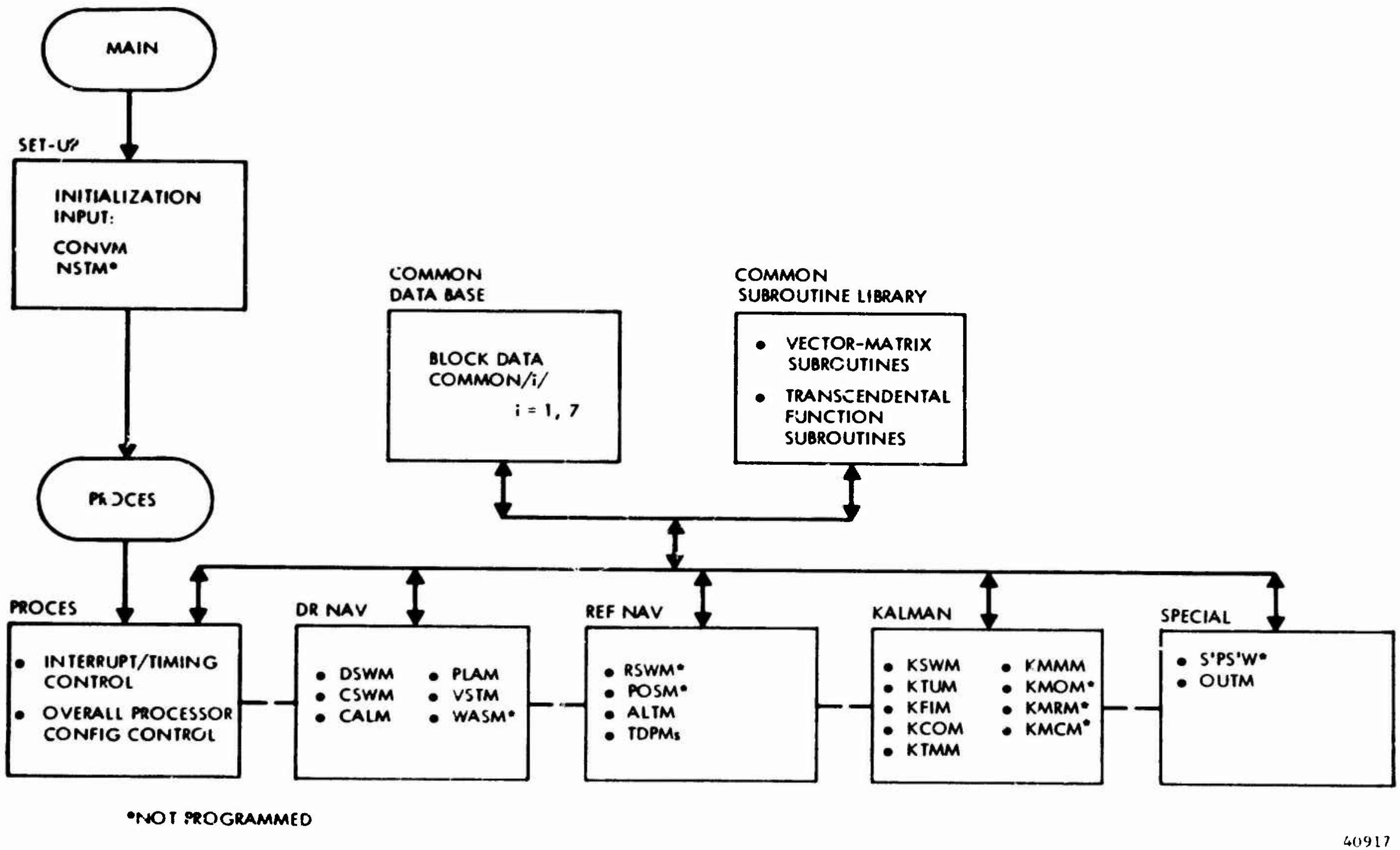


Figure 32. FORTRAN IV Processor MACRO Flow

TABLE LXXVIII. MAIN PROGRAM ORGANIZATION SUMMARY

```
                 ┌ BLOCK DATA (all common arrays, constants, etc.)
                 │    .
                 │    .
                 │    .
                 └   END

                 ┌ MAIN
                 │ CALL SET-UP (initialize variable arrays)
                 │ CALL CONVM  (input fixed constants)
MAIN PROGRAM     │ CALL NSTM   (input start-up condition of arrays and pointers)
                 │ CALL PROCES  (processor main loop timing/execution control)
                 │ CALL EXIT   (end of job)
                 │ RETURN
                 └ END

  ┌                SUBROUTINE PROCES
  │                    .
  │                    .
  │                    .
  │                  (labeled common arrays and variables)
  │                    .
  │                    .
  │                    .
  │              ┌ CALL DR NAV (initializes each D module to specified configuration)
  │    INIT.     │ CALL RF NAV (      "        R           "                     )
  │              │ CALL KALMAN (      "        K           "                     )
  │              └ CALL SPECAL (      "        O           "                     )
  │              ┌ CALL DSWM
  │              │ CALL CSWM
  │              │ CALL CALM
  │    DYN.      │ CALL PLAM
  │    NAV       │    .
  │    LOOP      │    .
  │              │    .
  │              │ CALL KMCM
  │              │ CALL OSWM
  └              └ CALL OUTM
```

The entire control program thus consists essentially of subroutine calls; each call (e.g., CALL PLAM) shifts program control to the execution group:

```
┌ ENTRY PLAM
│ (PLAM Module Operations)
└ RETURN
```

This arrangement efficiently allows for retention of module processing capability even in the absence of that module. A similar capability, using the FORTRAN GO TO, DO loop and CONTINUE operations, is embedded at the intramodular, specific algorithm level as well.

The remaining paragraphs of this subsection present the principal features and specifics of the FORTRAN IV program. It should be kept in mind in reviewing this discussion that almost all of these features and specifics are strongly influenced by the structure and characteristics of the higher order language (FORTRAN IV), the compiler (FORTRAN IV/IBM 370/65), and the machine (IBM 370/65) employed. The techniques selected therefore represent the best programming compromise in the programmer's judgment -- and against the background of the limited time allotted -- to realizing a program nucleus capable of generalization in any one of the several directions discussed earlier. (See the introduction to Section IV.)

b. Switching Modules*

This paragraph briefly describes and defines the markers used in the program to implement processor DR navigation mode, reference navigation configuration, and C frame switching requirements; i.e., the functions of the DSWM, CSWM, RSWM, and KSWM modules respectively. Descriptions of the functions of these modules has been omitted here, since these have already been presented in the corresponding MLI module specifications. The markers are summarized by module in Tables LXXIX through LXXXII.

TABLE LXXIX. DR NAV MODE SWITCHING MODULE (DSWM) MARKERS

Marker	Description	Marker Values	Meaning
DEQM DEQO	Current DR Equipment Availability Previous DR Equipment Availability	1 2 3 4 5	No IMU, AHRU or TAS AHRU and TAS only IMU only IMU and TAS only IMU, TAS and AHRU
DIMU	Current IMU Availability	1 2 3	No IMU IMU type (F) IMU type (S)
DIDR	IMU Status Control	1 2 3 4 5 6	Coarse Align Fine Align Alignment Complete IMU Nav Mode (IMU only) IMU & TAS only IMU, AHRU & TAS

*In addition to the required CSWM, this paragraph discusses the D, R, and K switching modules, which are not required for the limited, single-mode program scenario assumed here, but which have nevertheless been preliminarily formulated to provide a built-in base for later inclusion of full-mode switching capability.

TABLE LXXIX. (Continued)

Marker	Description	Marker Values	Meaning
DAHS	AHRU Availability	1 2 3	No AHRU AHRU DG Mode AHRU MAG Mode
DAHR	AHRU Status Control	1 2 3 4 5	No AHRU & TAS AHRU Align AHRU Align Complete AHRU DG Mode AHRU MAG Mode
DTAS	TAS Availability	1 2	No TAS TAS
DRMM DRMO	Current DR Nav Mode Marker Previous DR Nav Mode Marker	1 2 3 4 5	PDR Mode ADR Mode IDR (IMU only) IDR (IMU & TAS) IDR (IMU, TAS, AHRU)

TABLE LXXX. C FRAME SWITCHING MODULE (CSWM) MARKERS

Marker	Description	Marker Values	Meaning
CFRM	Current Nav Reference (C) Frame	1 2	No change Switch Initialization
KFFN	C Frame Switching Time Control	1 2 3	Do not Switch Complete KF Switching Kalman Switch Completed

TABLE LXXXI. REF NAV MEASUREMENT CONFIGURATION SWITCHING MODULE (RSWM) MARKERS

Marker	Description	Marker Values	Meaning
EBAR	Altitude Control	1 2 3 4 5 6	No Baro Altitude Pseudo Altitude Only Baro Altitude Field Altitude and Baro Altitude Field Altitude Pseudo and Field Altitude
DALT	Altitude Available		
ALMM	Altitude Mode		
REQM	Ref Nav Equipment Availability Marker	1 2	No Change in Equipment Availability Status Change in Equipment Availability Status

TABLE LXXXII. KALMAN FILTER SWITCHING MODULE (KSWM) MARKERS

Marker	Description	Marker Values	Meaning
KEQM KRMM REQM REQO	As Defined Above " " "		
KALL	Select KF Module Set to Execute		
NOMS	Number of Measurements to Process from Previous Cycle		
DSWTH	DR Nav - Switched	1 2	No Switch DR Switched
RSWTH	Ref Nav - Switched		
NOSW	Number of DR Nav Switches in Previous KF Interval		

c. Other Modules

The programming of almost all other (i.e., nonswitching) modules so closely follows the MLI specification as to be self-explanatory. No further definition is therefore given here.

d. Array Structure

The program extensively defines and utilizes FORTRAN data arrays to facilitate handling of the multidimensional data sets associated with (1) reference navigation measurement type availability, (2) selected measurement processing for the Kalman filter, and (3) Kalman filter estimate and covariance matrix processing. These are discussed in turn in the following paragraphs.

(1) Reference Navigation Measurement Type Availability Arrays

The one-dimensional array RKRAY(N) is used to define the availability and viability of reference navigation measurement equipment and output data. In this array the ranges N = 1 to 5, 6 to 10, 11 to 15, and 16 to 20 respectively contain reference altitude, reference ground LOS emitter, reference airborne emitter, and reference NAVSAT emitter data. In each such block of 5 words, the individual words represent the information summarized below (for each of the equipment group types just identified).

The first word of each block defines the current equipment status according to:

1 = Data input available from this net

2 = Add new emitter/input to this net

3 = Drop emitter/input from this net

4 = Drop complete net, discontinue processing data from this net

5 = Add a complete net.

Words 2 through 5 define emitter type and number according to:

1 = Emitter data available

2 = Add to net this emitter

3 = Drop emitter from net

4 = Spare

5 = Spare

(2) Measurement Processing Arrays

The 3-dimensional array PHI (L,M,N) is used here (a) to save data appropriate for synchronized measurement matrix generation at the end of the KF cycle, and (b) to save data appropriate for KF DR mode switching operations in the next KF cycle. The indices N, M, and L respectively denote:

N = Index of raw measurement data type.

M = Index of set of scalars, vectors, and matrices associated with use of a raw measurement data type N; i.e.,:

1. M_D	6. $\omega_{P/C}$	11. M_R
2. Y_D	7. $T_{P/C}$	12. Measurement type
3. Δt	8. C_D	13. Emitter number
4. Δv	9. Y_R	14. Clock (net)
5. Δp	10. C_R	15. Propagation data

L = index of elements of each of the entities of index M.

Another 3-dimensional processing array, DDM (L,M,N) is used for several purposes in the program, one of the more important of which is in the synchronized measurement time-smoothing operation. The indices M,N, and L here respectively denote:

M = index of raw measurement type; i.e.:

1. Position	4. Range rate (LOS)
2. Altitude	5. Range (EM)
3. Range (LOS)	

N = index of data group types associated with measurement type M; i.e.:

1. $\overline{Y}$, $\overline{Y}_D$, $\overline{Y}_R$, $\overline{C}_D$, $\overline{C}_R$, $\overline{M}_{Ds}$, $\overline{M}_{Rs}$
2. $\overline{N}_{Ds}$
3. $\overline{Z}_{Ds}$
4. $\overline{W}_{Ds}$

L = index of data within group type N of measurement type M.

(3) Kalman Filter Arrays

Substate-partition-related arrays are used extensively to store the variable sets defined in the detailed KF structure tables of the MLI KF structure specification. Since these are directly patterned on the MLI specification, no further definition is given here.

e. Subroutines

Both common and special FORTRAN subroutines are employed extensively in the program.

The common subroutines consist almost exclusively of vector-matrix operations; a few of these are general (IBM 370/65, FORTRAN IVH extended) FORTRAN V library subroutines, but the main body are navigation-specialized, Northrop library subroutines. The common subroutines are summarized in Table LXXXIII.

The special subroutines, which were developed specifically for this program, are of two kinds: (1) those which were designed in conjunction with, and facilitate the use of, the special data arrays described above (e.g., the special subroutine MXMV used in the KF measurement processing modules), and (2) those which mechanize operations which are expected to be highly application-dependent (e.g., the VSTM gravity computation).

TABLE LXXXIII. COMMON AND MATHEMATICAL SUBROUTINES

Mnemonic	Description
DOT	Dot product of two vectors
ADOT	Angle between two vectors (in degrees)
ADOTR	Angle between two vectors (in radians)
FNORM	Normalize a vector
VNORM	Normalize a vector
VCROS	Vector cross product
VADD	Vector addition
VSUB	Vector subtraction
VSCL	Scalar times a vector
VCXR	Vector times a vector transpose (outer product)
VRCX	Vector transpose times a vector (inner product)
VMOV	Vector transfer
ADDM	General matrix addition*
SUBM	General matrix subtract*
MPYM	Matrix multiply general*
MTRA	Matrix transpose general*
VTRT	Matrix transpose times a vector $(3x3)^T$ $(3x1)$
MTRT	Matrix times matrix transpose $(3x3)$ $(3x3)^T$
VTRN	Matrix times a vector (3x3) (3x1)
MTRN	Matrix times a matrix (3x3) (3x3)
WCROS	Vector into rotation matrix
STRN	Rotation (of a vector) about a coordinate axis (3x1)
GTRN	Generates a rotation matrix about a line (3x3)
GTSN	Generates a rotation matrix in terms of successive rotations about 3 axes (3x3)
INVERT	Invert a matrix
MSCL	Scalar times a matrix
MGID	Generate identity matrix
SINE	Sine of an angle
COSINE	Cosine of an angle
SQRT	Square root

*Essentially unmodified FORTRAN library subroutines

f. FORTRAN Program Listing

```
C      HOL(FORTRAN 4 ) PROCESSOR VERSION                                  MAIN     1
C                                                                         MAIN     2
C                                                                         MAIN     3
C                                                                         MAIN     4
C                                                                         MAIN     5
C      START OF PROGRAM                                                   MAIN     6
       CALL SETUP                                                         MAIN     7
       CALL CONVM                                                         MAIN     8
       CALL NSTM                                                          MAIN     9
       CALL PROCES                                                        MAIN    10
       CALL EXIT                                                          MAIN    11
       RETURN                                                             MAIN    12
       END                                                                MAIN    13
C      END OF MAIN PROGRAM                                                MAIN    14
C                                                                         MAIN    15
C                                                                         MAIN    16
C                                                                         MAIN    17
C      BLOCK DATA                                                         BLKD     1
       BLOCK DATA                                                         BLKD     2
       COMMON /A1/                                                        BLKD     3
     1 Y(3),F(3),VW(3),G(3),FP(3),PE(3),CE(3),GE(3),PS(3),P(3),DF(3)      BLKD     4
     2,DW(3),VAS(3),H,DV(3),DP(3),V(3),U1(3),U2(3),U3(3),CEN(3),DCSW(3)   BLKD     5
     3,TCEN(9),FACC(3),VASA(3)                                            BLKD     6
       COMMON /A11/ VEC(3),VEC1(3),VEC2(3),VEC11(9)                       BLKD     7
     2,VEC3(9),VEC4(9),VEC5(9),VEC6(9),VEC7(9),VEC8(9),VEC9(9),VEC10(9)   BLKD     8
     3,VEC12(36),VEC13(36),VEC14(36)                                      BLKD     9
       COMMON /A2/                                                        BLKD    10
     1 TAP(3,3),TLC(3,3),TPL(3,3),TPC(3,3),TCE(3,3),TAC(3,3),TAA(3,3)     BLKD    11
     2,TKA(3,3),TKL(3,3),TSW(3,3)                                         BLKD    12
       COMMON /A3/                                                        BLKD    13
     1 WLC(3),WPL(3),WPC(3),WEI(3),WAC(3),WAP(3),WPI(3),WE(3),WGYR(3)     BLKD    14
     2,WK(3)                                                              BLKD    15
       COMMON /A4/                                                        BLKD    16
```

```
1 DEOM,DEQO,DRMM,DRMO,DTAS,DAHR,DIMU,DIDR,EIMU,EAHR,EBAR              BLKD   17
2,CFRM,CFRO,CFCN,CFCO,DAHS,DALT,ALMM                                  BLKD   18
3,KEQM,KRMM,KRMO,REQM,REQO,KFFN                                       BLKD   19
4,     NOMES,DSWTH,RSWTH,NRSW,NOS1,NOMS,NOEM,NOSW                     BLKD   20
 DIMENSION TING(130)                                                  BLKD   21
 COMMON /A5/                                                          BLKD   22
1 FLTIME,DELT,STIME,BT1,TKAL,DELTP,DELRO,DELTC,DTPL,DELTV,FTKAL       BLKD   23
2,TDSU(10),TCSW(10),TCAL(10),TPAL(10),TWAS(10),TVST(10),TRSW(10)      BLKD   24
3,TALT(10),TPOS(10),TTDP(10),TKLT(10),TSPC(10)                        BLKD   25
4,BETA(3),DWK(3),DFK(3),PACC(9),PGCC(9),TPLF(9),TPLA(9)               BLKD   26
 COMMON/A51/ NO,SWTCH(100),RWTCH(100)                                 BLKD   27
 COMMON /A6/                                                          BLKD   28
1 VO(3),PO(3),VK(10),AMUR,DYMP                                        BLKD   29
2,HM(50),DMZ(10),DNS(50),DZS(50),XT(50)                               BLKD   30
3,UDDS(50),XDS(50),UDS(50),BBDM(50),DBDM(50),BKDM(50),UDM,QB          BLKD   31
 DIMENSION BRDM(37),BDRM(37),BBRM(37),DBRM(37),BKRM(37),RM(37)        BLKD   32
1,DRS(37),DRZ(37)                                                     BLKD   33
 DIMENSION DDM(50,6,4),OD11(6,6),OD13(6,3),GD13(6,3),XRS(37)          BLKD   34
1,URS(37),IKK(20)                                                     BLKD   35
 EQUIVALENCE (BBDM(13),BBRM(1)),(DBDM(13),DBRM(1)),(BKDM(13),BKRM(    BLKD   36
11)),(HM(13),RM(1)),(DNS(13),DRS(1)),(DZS(13),DRZ(1))                 BLKD   37
 EQUIVALENCE (XDS(13),XRS(1)),(UDS(13),URS(1))                        BLKD   38
 COMMON /A61/                                                         BLKD   39
1 PHI(50,15,10),DDMS(10,6,4),DMS(10,6,4),DMS1(10,6,4),DMR(20,6,4)     BLKD   40
2,RKARY(50),R(100),OD33(3,3),OD44(3,3)                                BLKD   41
3,GD33(3,3),TO11(3,3),TO12(3,3),TO21(3,3),TO22(3,3),OD131(3,3)        BLKD   42
4,OD132(3,3),GD131(3,3),GD132(3,3),AD44(9),AD4(3),OREJ(16)            BLKD   43
5,AREJ(16),ARUC,ARUA,OURA,OURC(4),OURE(16)                            BLKD   44
6,P11(6,6),P13(6,3),P14(6,3),P15(6,1),P16(6,2),P17(6,4)               BLKD   45
7,P31(3,6),P33(3,3),P34(3,3),P35(3,1),P36(3,2),P37(3,4)               BLKD   46
8,P41(3,6),P43(3,3),P44(3,3),P45(3,1),P46(3,2),P47(3,4)               BLKD   47
9,P51(1,6),P53(1,3),P54(1,3),P55(1,1),P56(1,2),P57(1,4)               BLKD   48
A,P61(2,6),P63(2,3),P64(2,3),P65(2,1),P66(2,2),P67(2,4)               BLKD   49
B,P71(4,6),P73(4,3),P74(4,3),P75(4,1),P76(4,2),P77(4,4)               BLKD   50
 EQUIVALENCE (OD11(1,1),TO11(1,1)),(OD11(10,1),TO12(1,1))             BLKD   51
```

```
1,(OD11(19,1),TO21(1,1)),(OD11(28,1),TO22(1,1))                      BLKD   52
2,(OD13(1,1),OD131(1,1)),(OD13(10,1),OD132(1,1))                     BLKD   53
3,(GD13(1,1),GD131(1,1)),(GD13(10,1),GD132(1,1))                     BLKD   54
4,(DDM(1,1,1),DDMS(1,1,1)),(DDM(241,1,1),DMS(1,1,1))                 BLKD   55
5,(DDM(481,1,1),DMS1(1,1,1)),(DDM(721,1,1),DMR(1,1,1))               BLKD   56
 COMMON /A7/                                                         BLKD   57
1 HB,HF,HR,HCR,DHBK,HK,HBC,HPC,SMALT,SIGB,TAUHB,SIGBW,HE,HBABA,HI    BLKD   58
2,ALTIM(100),ALAT(8),ALONG(8),HT(8),ALIM(8),RADIUS(5),ECCENT(5)      BLKD   59
3,INCLIN(5),ASCEND(5),PERIGE(5),TIME(5),ANGSEE(5),ANGROT(5)          BLKD   60
4,TAVANT(5),NA,NB                                                    BLKD   61
 COMMON /A71/RJ(3),UJ(3),D(3),DD(3),DEM(3),DEMD(3),RJD(3),DUD(3)     BLKD   62
1,ER,EJD(3),RMJ,RMJD,DRJ,DRJD,DTU,DTJ,ERD,YRJ,YRRJ,RMDP(3),RMDE(3)   BLKD   63
2,RMDTU(2),RMDTJ(2),RMDL,RRDPD(3),RRDED(3),RRTUD,RRTJD,RRDP(3)       BLKD   64
3,RRDE(3),DPTJ,DPIJ,EJ(3),CDJ,CRJ,YDJ,OK1,OK2,NC,IRT                 BLKD   65
 COMMON /A9/RO,DTR,CKFT,H2S,RTD,GZERO,RTS,WES,ROINV,VLITE            BLKD   66
1,PI,GOROSQ                                                          BLKD   67
 EQUIVALENCE (TING(1),TDSU(1)),(TING(11),TCSW(1))                    BLKD   68
1,(TING(21),TCAL(1)),(TING(31),TPAL(1)),(TING(41),TWAS(1))           BLKD   69
2,(TING(51),TVST(1)),(TING(61),TRSW(1)),(TING(71),TALT(1))           BLKD   70
3,(TING(81),TPOS(1)),(TING(91),TTDP(1))                              BLKD   71
4,(TING(101),TKLT(1)),(TING(111),TSPC(1))                            BLKD   72
 INTEGER    DEQM,DEQO,DRMM,DRMO,DTAS,DAHR,DIMU,DIDR,EIMU,EAHR,       BLKD   73
1EBAR,CFRM,CFRO,CFCN,CFCO,REQM,REQO,DAHS,ALMM                        BLKD   74
2,     NOMES,DSWTH,RSWTH,NRSW,NOS1,NOMS,NOEM,NOSW                    BLKD   75
 DATA TING/130*1.0/                                                  BLKD   76
 DATA FLTIME,TIME,DELT,STIME,BT1,DELTV /0.0,                         BLKD   77
1 0.0,0.1,10.,1.,.1/                                                 BLKD   78
 DATA BETA,DELTP,WE,DELRO,DELTC                                      BLKD   79
1,DWK,DFK,PACC,PGCC,TPLF,TPLA,DTPL /3*1.0,0.1,7.292116E-5,0.,0.,     BLKD   80
22.E+3,.1,3*.1,3*.1,9*0.,9*0.,1.,3*0.,1.,3*0.,1.,1.,3*0.,1.          BLKD   81
1,3*0.,1.,.1/                                                        BLKD   82
 DATA ALMM,EBAR,SMALT,SIGB,TAUHB,SIGBW,HBABA,HE,HI,ALTIM             BLKD   83
1 / 3,3,0.0,200.,14400.,10.,0.,0.,2.0E4,100*2.0E58/                  BLKD   84
 DATA DRMM,DRMO,DIDR,DAHR,DAHS,DTAS,DIMU,DALT/5,5,6,4,2,2            BLKD   85
```

```
      1,2,2./
       DATA RO,DTR,CKFT,H2S,RTD,GZERO,RTS,WES,ROINV,VLITE,PI,GOROSQ/
      1 2.0926E7,.01745329,1.687806,3600.,57.29578,32.0882,4.848137E-6
      2,7.29211E-5,4.784689E-8,47.08032,3.1415926,140127.2E10/
       DATA TAP,TLC,TPL,TPC,TCE,TAC,TAA,TKL,TKA,TSW,FACC/
      1 9*0.,9*0.,9*0.,9*0.,9*0.,9*0.,9*0.,9*0.,9*0.,9*0.,3*0.0/
       END
C
C
C
C
C      START PROGRAM SETUP
       SUBROUTINE SETUP
C      (TO RUN PROGRAM,COMMON BLOCK DATA AT END OF THIS LISTING
C      MUST BE INSERTED)
       REAL*8 DATR
       DIMENSION TITLE (20),NST(80),DEQP(10),REQP(10)
       NAMELIST /SWITCH/ NO,SWTCH
       NAMELIST /TIMING/TDSW,TCSW,TCAL,TPAL,TWAS,TVST,TRSW,TALT,TPOS
      1,TTDP,TKLT,TSPC,BT1,FLTIME,DELT,STIME,TKAL
       NAMELIST/CONST/BETA,DELTP,WE,DELRO,DELTC,DELTV
      1,DWK,DFK,PACC,PGCC,TPLF,TPLA,DTPL
       NAMELIST/NAEQ/ DRMM,DRMO,DIDR,DAHR,DAHS,DTAS,DIMU
      1,DALT,EBAR,NST
       NAMELIST /LOS/NB,ALAT,ALONG,HT,ANGSEE
       NAMELIST/NAV/RADIUS,ECCENT,INCLIN,ASCEND,PERIGE,TIME
      1,NA,ANGSEE,ANGROT,TAVANT
       DO 988 I=1,79
  988  Y(I)=0.0
       DO 987 I=1,30
  987  WLC(I)=0.0
       DO 51 I=1,157
```

```
      ALTIM(I)=0.0                                          STAU
   51 CONTINUE                                              STAU
   10 FORMAT (20A4)                                         STAU
   11 FORMAT (1H1,10X,20A4,25X,'DATE',2X,A8)                STAU
      CALL DATE(DATR)                                       STAU
      WRITE (6,9999)                                        STAU
 9999 FORMAT(2X,///,10X,'INPUT DATA',2X)                    STAU
      WE(1) = 7.292116E-5                                   STAU
      READ (5,10) TITLE                                     STAU
      WRITE (6,11) TITLE,DATR                               STAU
      READ  (5,NAEQ)                                        STAU
      DO 20 K =1,5                                          STAU
      IF (NST(K).EQ.0) GO TO 20                             STAU
      GO TO (46,47,48,49,50),K                              STAU
C     INPUT NAV.EQUIP.TIMING                                STAU
   46 CONTINUE                                              STAU
      GO TO 21                                              STAU
C     INPUT REF.EQUIP.TIMING                                STAU
   47 CONTINUE                                              STAU
      GO TO 21                                              STAU
C     INPUT MODULE TIMING                                   STAU
   48 CONTINUE                                              STAU
      READ (5,TIMING)                                       STAU
      DO 30 I=1,120                                         STAU
      TING(I) = BT1/TING(I)                                 STAU
   30 CONTINUE                                              STAU
      GO TO 21                                              STAU
   49 CONTINUE                                              STAU
      READ (5,LOS)                                          STAU
      WRITE (6,LOS)                                         STAU
      GO TO 21                                              STAU
```

```
 50   CONTINUF                                                    STAU  57
      READ (5,NAV)                                                STAU  58
      WRITE (6,NAV)                                               STAU  59
      GO TO 21                                                    STAU  60
 21   CONTINUE                                                    STAU  61
 20   CONTINUE                                                    STAU  62
      DEQM =1                                                     STAU  63
      IF (DIMU.EQ.2.OR.DIMU.EQ.3) DEQM=3                          STAU  64
      IF (DTAS.EQ.2) DEQM=DEQM+1                                  STAU  65
      IF (DAHS.EQ.2.OR.DAHS.EQ.3) GO TO 23                        STAU  66
      GO TO 24                                                    STAU  67
 23   GO TO (27,27,26,26),DEQM                                    STAU  68
 26   DEQM = 5                                                    STAU  69
 27   CONTINUE                                                    STAU  70
 24   CONTINUE                                                    STAU  71
      RETURN                                                      STAU  72
C     FND PROGRAM SETUP                                           STAU  73
C                                                                 CONM   1
C                                                                 CONM   2
C                                                                 CONM   3
C                                                                 CONM   4
C     START PROCESSOR SETUP                                       CONM   5
C                                                                 CONM   6
C     START OF CONM (SUBROUTINE)                                  CONM   7
      ENTRY CONVM                                                 CONM   8
      READ (5,CONST)                                              CONM   9
      RETURN                                                      CONM  10
C     END OF CONM (SUBROUTINE)                                    CONM  11
C     START OF NSTM (SUBROUTINE)                                  NSTM   1
      ENTRY NSTM                                                  NSTM   2
C      ARRAY PARAMETERS (I),I=1,N)                                NSTM   3
C     SWITCH (I) SWITCH TIME                                      NSTM   4
C                TYPE (1=C-EF,2=EF-C,3=REF.NAV ON)                NSTM   5
C                COOR.TYPE(1,LAT,LONG,2=POLAR,3=X,Y,Z,4=UTM)      NSTM   6
C                LATITUDE  DEGREES                                NSTM   7
C                LONGITUDE DEGREES                                NSTM   8
C                THETA     DEGREES                                NSTM   9
C                ALTITUDE  FEET                                   NSTM  10
```

```
C                *                                                     NSTM   11
C                *                                                     NSTM   12
C                *                                                     NSTM   13
C                *                                                     NSTM   14
      READ (5,SWITCH)                                                  NSTM   15
      READ (5,RWTCH)                                                   NSTM   16
      WRITE (6.RWTCH)                                                  NSTM   17
      RETURN                                                           NSTM   32
C     END OF NSTM (SUBROUTINE)                                         NSTM   33
C     END PROCESSOR SETUP                                              NSTM   34
      END                                                              NSTM   35
C                                                                      PROC    1
C                                                                      PROC    2
C                                                                      PROC    3
C     START PROCESSOR EXECUTION CONFIGURATION SETUP                    PROC    4
      SUBROUTINE PROCES                                                PROC    5
C     (TO RUN PROGRAM,COMMON BLOCK DATA AT END OF THIS LISTING         PROC    6
C     MUST BE INSERTED)                                                PROC    7
      DIMENSION TV(20),TO(20)                                          PROC    8
      CALL DRNAV                                                       PROC    9
      CALL RFNAV                                                       PROC   10
      CALL KALMAN                                                      PROC   11
      CALL SPECL                                                       PROC   12
      DO 20 I=1,20                                                     PROC   13
      TO(I)=0.                                                         PROC   14
 20   TV(I) = 0.0                                                      PROC   15
C     END PROCESSOR EXECUTION CONFIGURATION SETUP                      PROC   16
C                                                                      PROC   17
C                                                                      PROC   18
C                                                                      PROC   19
C                                                                      PROC   20
C     BEGINNING OF PROCESSOR DYNAMIC LOOP                              PROC   21
 23   K = 1                                                            PROC   22
      IF (FLTIME.GT.STIME) GO TO 25                                    PROC   23
      DO 21 I1 = 1,120,10                                              PROC   24
      IF (TV(K).GT.FLTIME) GO TO 22                                    PROC   25
      TV(K) = TING(I1) + FLTIME                                        PROC   26
```

```
C
C      DR NAV SUBMODULE CALLS
       GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13),K
 1     CONTINUE
       CALL DSWM
       GO TO 22
 2     CONTINUE
       CALL CSWM
       GO TO 22
 3     CONTINUE
       CALL CALM
       GO TO 22
 4     CONTINUE
       CALL PLAM
       GO TO 22
 5     CONTINUE
       CALL WASM
       GO TO 22
 6     CONTINUE
       CALL VSTM
       GO TO 22
C
C      REF NAV SUBMODULE CALLS
 7     CONTINUE
       CALL RSWM
       GO TO 22
 8     CONTINUE
       CALL ALTM
       GO TO 22
 9     CONTINUE
       CALL POSM
       GO TO 22
 10    CONTINUE
       CALL TDPM
       GO TO 22
C
C      KALMAN FILTER MODULE SUBROUTINE CALLS
 11    CONTINUE
       CALL KALMN
```

```
      GO TO 22                                                          PROC  66
 12   CONTINUE                                                          PROC  67
C                                                                       PROC  68
C     SPECIAL MODULE SUBROUTINE CALLS                                   PROC  69
      CALL OUTM                                                         PROC  70
 22   K= K + 1                                                          PROC  71
 13   CONTINUE                                                          PROC  72
 21   CONTINUE                                                          PROC  73
      FLTIME = FLTIME + DELT                                            PROC  74
      GO TO 23                                                          PROC  75
 25   CONTINUE                                                          PROC  76
      RETURN                                                            PROC  77
      END                                                               PROC  78
C     END OF PROCESSOR DYNAMIC LOOP                                     PROC  79
C     END OF SUBROUTINE PROCES                                          PROC  80
C                                                                       DRNV   1
C                                                                       DRNV   2
C                                                                       DRNV   3
C                                                                       DRNV   4
C     MODULE SUBROUTINES                                                DRNV   5
C                                                                       DRNV   6
C                                                                       DRNV   7
C     DR NAV MODULE GROUP                                               DRNV   8
C     START OF SUBROUTINE DR-NAV                                        DRNV   9
      SUBROUTINE DRNAV                                                  DRNV  10
C     (TO RUN PROGRAM,COMMON BLOCK DATA AT END OF THIS LISTING          DRNV  11
C     MUST BE INSERTED)                                                 DRNV  12
      DIMENSION TVC(10),TVP(10),TVW(10),TVV(10)                         DRNV  13
      NET = 1                                                           DRNV  14
      DO 9 I = 1,10                                                     DRNV  15
      TVC(I) = 0.0                                                      DRNV  16
      TVP(I) = 0.0                                                      DRNV  17
      TVW(I) = 0.0                                                      DRNV  18
      TVV(I) = 0.0                                                      DRNV  19
  9   CONTINUE                                                          DRNV  20
      CFRM = 1                                                          DRNV  21
      KFFN = 2                                                          DRNV  22
      RETURN                                                            DRNV  23
```

```
C       START OF DSWM (SUBROUTINE)                                  DSWM
        ENTRY DSWM                                                  DSWM
        IF (DEQM.EQ.DEQO.AND.DRMM.EQ.DRMO) RETURN                   DSWM
        GO TO (10,11,12,13,14),DEQM                                 DSWM
C        PDR MODE (NO IMU,TAS,AND AHRU)                             DSWM
   10   IF (DRMM.EQ.1.AND.DRMO.EQ.1) GO TO 16                       DSWM
C        SET MARKERS FOR PDR, INITIALIZE SUBMODULES TO PDR          DSWM
C       PDR ONLY                                                    DSWM
  16    CONTINUE                                                    DSWM
        DRMM=1                                                      DSWM
        DRMO = 1                                                    DSWM
        DEQM = 1                                                    DSWM
        DEQO = 1                                                    DSWM
        DIDR = 1                                                    DSWM
        GO TO 1                                                     DSWM
C        ADR MODE (AHRU AND TAS)                                    DSWM
  11    IF (DRMM.EQ.2.AND.DRMO.EQ.2) GO TO 17                       DSWM
C       AHRU VALIDITY DETERMINE                                     DSWM
  30    GO TO (33,34,35,17,17),DAHR                                 DSWM
  33    DAHR = 2                                                    DSWM
  34    RETURN                                                      DSWM
C       SET MARKERS FOR ADR,INITIALIZE SUBMODULES TO ADR            DSWM
  35    DAHR = DAHR + EAHR                                          DSWM
        DTAS = 2                                                    DSWM
  17    CONTINUE                                                    DSWM
        DRMM=2                                                      DSWM
        DRMO = 2                                                    DSWM
        DEQM = 2                                                    DSWM
        DEQO = 2                                                    DSWM
        DIDR = 1                                                    DSWM
        GO TO 1                                                     DSWM
C       IDR MODE (IMU,NO TAS OR AHRU)                               DSWM
  12    IF (DRMM.EQ.3.AND.DRMO.EQ.3) GO TO 18                       DSWM
  32    CONTINUE                                                    DSWM
        GO TO (36,37,18,18,18,18),DIDR                              DSWM
  36    DIDR = 2                                                    DSWM
  37    CONTINUE                                                    DSWM
        GO TO (41,42,43,44,44),DAHR                                 DSWM
```

```
C       PDR UNTIL AHRU OR IMU VALID                               DSWM  39
 41     DAHR = 2                                                  DSWM  40
 42     DRMM = 1                                                  DSWM  41
        GO TO 1                                                   DSWM  42
 43     DAHR = DAHR + EAHR                                        DSWM  43
C       ADR UNTIL IMU VALID                                       DSWM  44
        DRMM = 2                                                  DSWM  45
 44     GO TO 1                                                   DSWM  46
C       SET MARKERS FOR IDR MODE                                  DSWM  47
C       INITIALIZE SUBMODULES TO IDR                              DSWM  48
 18     DTAS = 1                                                  DSWM  49
        DAHR = 1                                                  DSWM  50
        DRMM = 3                                                  DSWM  51
        DRMO = 3                                                  DSWM  52
        DEOM = 3                                                  DSWM  53
        DEOO = 3                                                  DSWM  54
        DIDR = 4                                                  DSWM  55
        GO TO 1                                                   DSWM  56
C      IDR MODE                                                   DSWM  57
 13     IF (DRMM.EQ.4.AND.DRMO.EQ.4) GO TO 19                     DSWM  58
        IF (DIDR.LT.3) GO TO 32                                   DSWM  59
C        SET IDR MODE,INITIALIZE TO IDR MODE                      DSWM  60
 19     CONTINUE                                                  DSWM  61
        DTAS=2                                                    DSWM  62
        DRMM = 4                                                  DSWM  63
        DRMO = 4                                                  DSWM  64
        DEOM = 4                                                  DSWM  65
        DEOO = 4                                                  DSWM  66
        DIDR = 5                                                  DSWM  67
        GC TO 1                                                   DSWM  68
C       IDR MODE,AHRU TAS,IMU                                     DSWM  69
 14     IF (DRMM.EQ.5.AND.DRMO.EQ.5) GO TO 20                     DSWM  70
        IF (DIDR.LT.3) GO TO 32                                   DSWM  71
C       STE IDR MODE,INITIALIZE MODULES TO IDR MODE               DSWM  72
 20     CONTINUF                                                  DSWM  73
        DRMM=5                                                    DSWM  74
        DRMO = 5                                                  DSWM  75
        DFOM = 5                                                  DSWM  76
        DEOO = 5                                                  DSWM  77
```

```
      DIOR = 6                                                          DSWM  78
      DTAS = 2                                                          DSWM  79
C     SET KALMAN SWITCH MARKER                                          DSWM  80
 1    KRMM = DRMM                                                       DSWM  81
      RETURN                                                            DSWM  82
C     END OF DSWM (SUBROUTINE)                                          DSWM  83
C                                                                       CSWM   1
C     START OF CSWM (SUBROUTINE)                                        CSWM   2
      ENTRY CSWM                                                        CSWM   3
      IF (CFRM.GT.1) GO TO 101                                          CSWM   4
 100  IF (FLTIME.LT.SWTCH(NET)) GO TO 99                                CSWM   5
      IN= SWTCH(NET+1)                                                  CSWM   6
      IS= SWTCH(NET+2)                                                  CSWM   7
      GO TO (110,111,112,113),IN                                        CSWM   8
 110  GO TO (114,114,114,114),IS                                        CSWM   9
C     LAT.,LONG. COOR. INPUT                                            CSWM  10
C     EF TO C SWITCH                                                    CSWM  11
 114  CONTINUE                                                          CSWM  12
      CALL VSCL (O,VEC,CEN)                                             CSWM  13
      CALL MGID(TCEN,4)                                                 CSWM  14
      GO TO 1011                                                        CSWM  15
 111  CONTINUE                                                          CSWM  16
C     C TO EF SWITCH                                                    CSWM  17
      SC = SIND (SWTCH(NET+3))                                          CSWM  18
      CC = COSD (SWTCH(NET+3))                                          CSWM  19
      SL = SIND (SWTCH(NET+4))                                          CSWM  20
      CL = COSD (SWTCH(NET+4))                                          CSWM  21
C     COMPUTE NEW C(C/E)                                                CSWM  22
      HTX=SWTCH(NET+6)+RO                                               CSWM  23
      VEC(1) = SC                                                       CSWM  24
      VEC(2) = CC*CL                                                    CSWM  25
      VEC(3)= CC*SL                                                     CSWM  26
      CALL VSCL(HTX,VEC,CEN)                                            CSWM  27
C     COMPUTE  NEW T(C/E)                                               CSWM  28
      CALL MSCL (O,VEC2,TCEN)                                           CSWM  29
      K = 1                                                             CSWM  30
      DO 1121 I=1,9,3                                                   CSWM  31
      TCEN(I) = VEC(K)                                                  CSWM  32
```

```
 1121 K=K+1                                         CSWM
      TCEN(3) = CC                                  CSWM
      TCEN(5) = -SL                                 CSWM
      TCEN(6) = -SC*CL                              CSWM
      TCEN(8) = CL                                  CSWM
      TCEN(9) = -SC*SL                              CSWM
C     COMPUTE(TSW)                                  CSWM
 1011 CONTINUE                                      CSWM
      CALL MTRT (TCEN,TCE,TSW)                      CSWM
      CALL VSUB (CEN,CE,DCSW)                       CSWM
      CFCN = 2                                      CSWM
      CFRM = 2                                      CSWM
      NET = NET+10                                  CSWM
      GO TO(101,101,103),IN                         CSWM
 103  CONTINUE                                      CSWM
      GO TO 100                                     CSWM
C     CHECK IN OF KALMAN CYCLE                      CSWM
 101  IF (KFFN.NE.2) GO TO 10C                      CSWM
C     SWITCH ALL NON-KALMAN                         CSWM
C     VARIABLES                                     CSWM
      CALL VMOV (CEN,CE,3)                          CSWM
      CALL MXMV (TCEN,TCE,9)                        CSWM
      CALL VTRN (TSW,P,VEC)                         CSWM
      CALL VTRN (TCEN,DCSW,VEC1)                    CSWM
      CALL VSUB (VEC,VEC1,P)                        CSWM
      CALL VTRN (TSW,V,VEC)                         CSWM
      CALL VMOV (VEC,V,3)                           CSWM
      CALL VTRN (TSW,G,VEC)                         CSWM
      CALL VMOV (VEC,G,3)                           CSWM
      CALL VTRN(TSW,DP,VEC)                         CSWM
      CALL VMOV(VEC,DP,3)                           CSWM
      CALL MTRN(TSW,TLC,VEC2)                       CSWM
      CALL MXMV(VEC2,TLC,9)                         CSWM
      IF (DRMM.LT.2) GO TO 120                      CSWM
C     EITHER (IDR OR ADR)                           CSWM
      CALL VTRN(TSW,WEI,VEC)                        CSWM
      CALL VMOV(VEC,WEI,3)                          CSWM
      CALL VTRN(TSW,WAC,VEC)                        CSWM
      CALL VMOV(VEC,WAC,3)                          CSWM
```

```
      CALL MTRN (TSW,TPC,VEC2)                                          CSWM  72
      CALL MXMV (VEC2,TPC,9)                                            CSWM  73
      CALL MTRN (TSW,TAC,VEC2)                                          CSWM  74
      CALL MXMV (VEC2,TAC,9)                                            CSWM  75
C     CHECK TAS AVAILABLE                                               CSWM  76
      IF (DTAS.NE.2) GO TO 121                                          CSWM  77
      CALL VTRN (TSW,VW,VEC)                                            CSWM  78
      CALL VMDV (VEC,VW,3)                                              CSWM  79
      CALL VTRN (TSW,VAS,VEC)                                           CSWM  80
      CALL VMOV (VEC,VAS,3)                                             CSWM  81
  121 IF (DRMM.EQ.2) GO TO 999                                          CSWM  82
  120 CALL VTRN (TSW,F,VEC)                                             CSWM  83
      CALL VMOV (VEC,F,3)                                               CSWM  84
      CALL VTRN(TSW,DV,VEC)                                             CSWM  85
      CALL VMOV(VEC,DV,3)                                               CSWM  86
      IF (DRMM.NE.1) GO TO 999                                          CSWM  87
      CALL VTRN(TSW,U1,VEC)                                             CSWM  88
      CALL VMOV (VEC,U1,3)                                              CSWM  89
      CALL VTRN (TSW,U2,VEC)                                            CSWM  90
      CALL VMOV (VEC,U2,3)                                              CSWM  91
      CALL VTRN(TSW,U3,VEC)                                             CSWM  92
      CALL VMOV (VEC,U3,3)                                              CSWM  93
  999 CONTINUE                                                          CSWM  94
C     SET MARKER FOR KALMAN AND REF.NAV.SW.                             CSWM  95
      KFFN = 3                                                          CSWM  96
      CFRM = 1                                                          CSWM  97
      GO TO 100                                                         CSWM  98
  113 CONTINUE                                                          CSWM  99
  112 CONTINUE                                                          CSWM 100
      NET =NET+10                                                       CSWM 101
      GO TO 100                                                         CSWM 102
   99 CONTINUE                                                          CSWM 103
      RETURN                                                            CSWM 104
C     END OF CSWM (SUBROUTINE)                                          CSWM 105
C                                                                       CALM   1
C     START OF CALM (SUBROUTINE)                                        CALM   2
      ENTRY CALM                                                        CALM   3
      RETURN                                                            CALM   4
C     END OF CALM (SUBROUTINE)                                          CALM   5
C                                                                       CALM   6
```

```
C       START OF PLAM (SUBROUTINE)
        ENTRY PLAM
        IF (ORMM.EQ.1) RETURN
        DO 200 I = 2,10
        KR = I
        IF (TVP(KR).GT.FLTIME) GO TO 200
        TVP(KR) = TPAL(KR) + FLTIME
        GO TO (200,201,202,203,204,205,206,207,208,209),KR
C       COMPUTE (KL,WLC,TLC,AND WEI)
 201    CONTINUE
        GD = FNORM(PE)
        GF = 1./GD
        CALL VSCL (0,VEC1,VEC)
        VEC(1) = GF
        CALL VTRN(TLC,WE,WEI)
        CALL WCROS (VEC,TKL)
        CALL VTRN (TLC,V,VEC)
        CALL VTRN (TKL,VEC,WLC)
        CALL WCROS (WLC,VEC2)
        CALL MTRN  (VEC2,TLC,VEC3)
        CALL MSCL (DELTC,VEC3,VEC2)
        CALL SUBM (TLC,VEC2,VEC3,3,3)
        CALL MXMV (VEC3,TLC,9)
        GO TO 200
C       COMPUTE (WPI,WPL)
 202    CONTINUE
        IF (DRMM.EQ.2) GO TO 214
        IF (DIMU.EQ.3) GO TO 213
 215    CALL VMOV(WK,WPL,3)
        GO TO 200
C       PLATFORM TYPE (S)
 213    CONTINUE
        CALL VSUB(DW,DWK,VEC)
        CALL VSUB(WGYR,VEC,WPI)
        CALL VADD(WLC,WEI,VEC)
        CALL VTRN(TPL,VEC,VEC1)
        CALL VSUB(WPI,VEC1,WPL)
        GO TO 200
C       DRNAV IS (ADR)
```

```
 214  CONTINUE
      IF (DAHR.EQ.4) GO TO 215
C     AHRU IN MAG.MODE
      CALL VCROS(WEI,G,VEC)
      CALL VRXC (V,VEC,VEC1)
      GD = FNORM(G)
      CALL VSCL (GD,PE,VEC2)
      CALL VRXC (G,WEI,VEC3)
      CALL VCROS (VEC,WEI,VEC4)
      CALL VRXC (VEC2,VEC4,VEC)
      GF =(VEC1(1)* VEC3(1))/(VEC(1))
      CALL VNORM (G,VEC)
      CALL VSCL (GF,VEC,WPL)
      GO TO 200
C     COMPUTE (WPL,TPL,WPC,TPC)
 203  CONTINUE
      CALL WCROS (WPL,VEC2)
      CALL MTRN  (VEC2,TPL,VEC3)
      CALL MSCL  (DTPL,VEC3,VEC2)
      CALL SUBM(TPL,VEC2,VEC3,3,3)
      CALL MXMV  (VEC3,TPL,9)
      CALL VTRN  (TPL,WLC,VEC)
      CALL VADD  (WPL,VEC,WPC)
      CALL MTRN(TPL,TLC,TPC)
      GO TO 200
C     COMPUTE SPECIFIC FORCE
 204  CONTINUE
C     DF AND DW COMPUTATION ARE NOT MECHANIZED
      IF (DRMM.EQ.2) GO TO 200
      IF (DIMU.EQ.3) GO TO 210
C     PLATFORM TYPE (F)
      CALL VTRN(TPL,WEI,VEC)
      CALL VADD(DWK,VEC,VEC1)
      CALL VADD(VEC1,WPC,WPI)
      CALL VADD(WPI,DW,WGYR)
      CALL VADD (DF,DFK,VEC)
      CALL VADD (FACC,VEC,FP)
      GO TO 200
```

```
C       PLATFORM TYPE (S)                                                PLAM   78
  210   CONTINUE                                                         PLAM   79
        GO TO 200                                                        PLAM   80
C       COMPUTE  (TAP,WAP)                                               PLAM   81
  205   CONTINUE                                                         PLAM   82
        GO TO 200                                                        PLAM   83
C       COMPUTE  (TAC)                                                   PLAM   84
  206   CONTINUE                                                         PLAM   85
C       COMPUTE (TAC,WAC)                                                PLAM   86
        CALL MTRN (TAP,TPC,TAC)                                          PLAM   87
        CALL VADD(WAP,WPC,VEC)                                           PLAM   88
        CALL VTRT(TPC,VEC,WAC)                                           PLAM   89
  207   CONTINUE                                                         PLAM   90
  208   CONTINUE                                                         PLAM   91
  209   CONTINUE                                                         PLAM   92
  200   CONTINUE                                                         PLAM   93
        RETURN                                                           PLAM   94
C       END OF PLAM (SUBROUTINE)                                         PLAM   95
C                                                                        WASM    1
C       START OF WASM (SUBROUTINE)                                       WASM    2
        ENTRY WASM                                                       WASM    3
        RETURN                                                           WASM    4
C       END OF WASM (SUBROUTINE)                                         WASM    5
C                                                                        VSTM    1
C       START OF VSTM (SUBROUTINE)                                       VSTM    2
        ENTRY VSTM                                                       VSTM    3
        DO 150 I=2,10                                                    VSTM    4
        KR = I                                                           VSTM    5
        IF (TVV(KR).GT.FLTIME) GO TO 150                                 VSTM    6
        TVV(KR) = TVST(KR) + FLTIME                                      VSTM    7
C                    F    V    P    PE   GE   G   /H/ WEI                VSTM    8
        GO TO (150,151,153,154,155,156,157,158,152,150),KR               VSTM    9
C       COMPUTE SPECIFIC FORCE (F)                                       VSTM   10
   151  CONTINUE                                                         VSTM   11
        IF (DRMM.GE.3) GO TO 160                                         VSTM   12
        IF (DRMM.EQ.2) GO TO 150                                         VSTM   13
C       MODE IS PDR (COMPUTE C FRAME REF. ACCEL'N IN L FRAME             VSTM   14
C       AND L/C FRAME TRANSFORMATION T  L/C                              VSTM   15
        GO TO 150                                                        VSTM   16
```

```
C     MODE = IDR
 160  CONTINUE
      CALL VTRT (TPC,FP,F)
      GO TO 150
 153  CONTINUE
C     COMPUTE VELOCITY (V)
      IF (DRMM.EQ.2) GO TO 161
      IF (DRMM.EQ.1) GO TO 1611
C     MODE IDR
      CALL VSCL(2.0,WEI,VEC)
      CALL VCROS(VEC,V,VEC1)
      CALL VSUB (G,VEC1,VEC)
      CALL VADD (F,VEC,VEC1)
      CALL VSCL (DELTV,VEC1,DV)
      CALL VADD(V,DV,V)
      GO TO 150
C     ADR VELOCITY (V)
 161  CONTINUE
      CALL VADD (VW,VAS,VEC)
      CALL VSUB (VEC,V,DV)
      CALL VMOV (VEC,V,3)
      GO TO 150
 1611 CONTINUE
C     MODE PDR COMPUTE VELOCITY
      GO TO 150
C     POSITION (P)
 154  CONTINUE
      CALL VSCL (DELTP,V,DP)
      CALL VADD(P,DP,P)
      GO TO 150
C     POSITION (PE)
 155  CONTINUE
      CALL VTRT (TCE,P,VEC)
      CALL VADD (VEC,CE,PE)
      GO TO 150
C     GRAVITY (GE)
 156  GD = FNORM(PE)
      IF (GD.LE.(RO-DELRO)) GO TO 162
```

```
      GC = FNORM(WEI)
C     NEED SCALER GENERATION HERE
      GF = (GOROSQ/((GD**2)*GD)) -(GC**2)
      CALL MGID(VEC2,4)
      CALL MSCL(GF,VEC2,VEC3)
      CALL VCXR(WE,WE,VEC2)
      CALL ADDM (VEC2,VEC3,VEC4,3,3)
      CALL MSCL(-1.0,VEC4,VEC3)
      CALL VTRN(VEC3,PE,GE)
      GO TO 150
  162 CONTINUE
      CALL VSCL (0,VEC,GE)
      GO TO 150
C     COMPUTE GRAVITY (G)
  157 CONTINUE
      CALL VTRN (TCE,GE,G)
      GO TO 150
C     COMPUTE ALTITUDE (/H/)
  158 CONTINUE
      GC =FNORM(PE)
      GF = RO/GC
      CALL VSCL (GF,PE,PS)
      GF = FNORM(PS)
      H = ABS(GC - GF)
      GO TO 150
  152 CONTINUE
  150 CONTINUE
      RETURN
      END
C     END OF VSTM (SUBROUTINE)
C     END OF SUBROUTINE DR-NAV

C
C
C     REF-NAV MODULE GROUP
      SUBROUTINE RFNAV
C     (TO RUN PROGRAM,COMMON BLOCK DATA AT END OF THIS LISTING
C     MUST BE INSERTED)
      DIMENSION TN(3,3,8),TR(3,3),TP'5),EN(5),RMN(5),ECC(5),ERS(5)
```

```
     1.PP(5.3).QQ(5.3)                                                    RFNV   9
C     DIMENSION TQ(20)                                                    RFNV  10
      NAMELIST /NALTD/EBAR,HI,SMALT,NO,SIGB,TAUHB,SIGBW,HBABA,HE,ALTIM    RFNV  11
      NY=1                                                                RFNV  12
      HB = 0.                                                             RFNV  13
      HF = 0.                                                             RFNV  14
      HBARO = 0.                                                          RFNV  15
      IF (DALT.EQ.0.0) GO TO 10                                           RFNV  16
C     INPUT ALTITUDE PARAMETERS                                           RFNV  17
      READ (5,NALTD)                                                      RFNV  18
 10   CONTINUE                                                            RFNV  19
      ALMM = EBAR                                                         RFNV  20
      SS = TKAL                                                           RFNV  21
      IF (SMALT.EQ.0.) SMALT = TKAL                                       RFNV  22
      NL = 1                                                              RFNV  23
      HPC = 0.                                                            RFNV  30
      HCR = 0.                                                            RFNV  31
      DHBK= 0.                                                            RFNV  32
      RETURN                                                              RFNV  33
C                                                                         RSWM   1
C     START OF RSWM (SUBROUTINE)                                          RSWM   2
      ENTRY RSWM                                                          RSWM   3
C     EMITTER SWITCHING                                                   RSWM   4
C      RWTCH ARRAY (1)= TIME                                              RSWM   5
C                    2 =EMITTER NET TYPE  1=GR.,2=AIRBORNE,3=NAVSTAT      RSWM   6
C                 3- 6 = EMITTER NO.OR NET INFC                           RSWM   7
C                    7 = TIME  (ETC.)                                     RSWM   8
C                                                                         RSWM   9
C                                                                         RSWM  10
 30   CONTINUE                                                            RSWM  11
      IF (FLTIME.LT.RWTCH(NY)) GO TO 31                                   RSWM  12
      NZ=NY+1                                                             RSWM  13
      NX=RWTCH(NZ)                                                        RSWM  14
      IF (NX.GT.3) GO TO 32                                               RSWM  15
      NP=2                                                                RSWM  16
 33   CONTINUE                                                            RSWM  17
      NT=NZ+1                                                             RSWM  18
      NTT=NT+3                                                            RSWM  19
      NR=NX*5+2                                                           RSWM  20
```

```
      RKARY(NR-1)=NP                                                    RSWM  21
      DO 34 I=NT,NTT                                                    RSWM  22
      RKARY(NR)=RWTCH(I)                                                RSWM  23
      NR=NR+1                                                           RSWM  24
   34 CONTINUE                                                          RSWM  25
      GO TO 35                                                          RSWM  26
   32 CONTINUE                                                          RSWM  27
      IF (NX.GT.6) GO TO 36                                             RSWM  28
      NP=5                                                              RSWM  29
      GO TO 33                                                          RSWM  30
   36 CONTINUE                                                          RSWM  31
      NP=4                                                              RSWM  32
      GO TO 33                                                          RSWM  33
   35 CONTINUE                                                          RSWM  34
      REQM=2                                                            RSWM  35
      NY=NY+6                                                           RSWM  36
      GO TO 30                                                          RSWM  37
   31 CONTINUE                                                          RSWM  38
      RETURN                                                            RSWM  39
C     END OF RSWM (SUBROUTINE)                                          RSWM  40
C                                                                       ALTM   1
C     START OF ALTM (SUBROUTINE)                                        ALTM   2
      ENTRY ALTM                                                        ALTM   3
      IF (FLTIME.LT.CALT) RETURN                                        ALTM   4
      CALT=FLTIME + SMALT                                               ALTM   5
      IF (FLTIME.LE.ALTIM(NL)) GO TO 11                                 ALTM   6
      ALMM = ALTIM(NL+1)                                                ALTM   7
      HI = ALTIM(NL+2)                                                  ALTM   8
      IF (ALMM.EQ.1) RETURN                                             ALTM  10
      EBAR = ALMM                                                       ALTM  11
   11 CONTINUE                                                          ALTM  14
C     CHECK ALTITUDE MODE                                               ALTM  15
      GO TO (12,12,13,14,15,15,15,15),ALMM                              ALTM  16
C     MODE = APS                                                        ALTM  17
   12 CONTINUE                                                          ALTM  18
C     MODE = AB                                                         ALTM  19
   13 CONTINUE                                                          ALTM  20
C     MODE = GFB                                                        ALTM  21
```

```
 14   CONTINUE                                          ALTM   22
C     MODE = GF                                         ALTM   23
 15   CONTINUE                                          ALTM   24
      HR = HBARO                                        ALTM   26
      HR = H - HBARO                                    ALTM   34
      IF (AIR.EQ.0.0) RETURN                            ALTM   35
C     AIRBORNE                                          ALTM   36
      GO TO (16,16,17,18,19,19,19,19),ALMM              ALTM   37
C     MODE = APS                                        ALTM   38
 16   CONTINUE                                          ALTM   39
      HPC = HPC*HK*(HPC-HCR)                            ALTM   40
      GO TO 20                                          ALTM   41
C     MODE = AB                                         ALTM   42
 17   CONTINUE                                          ALTM   43
      HBC = HB + DHBK                                   ALTM   44
      GO TO 20                                          ALTM   45
C     MODE = GFB                                        ALTM   46
 18   CONTINUE                                          ALTM   47
      HBC = HF                                          ALTM   48
      DHBK = HF - HB                                    ALTM   49
      GO TO 20                                          ALTM   50
C     MODE = GF                                         ALTM   51
 19   CONTINUE                                          ALTM   52
      HPC = HF                                          ALTM   53
 20   CONTINUE                                          ALTM   54
      HRC = HR                                          ALTM   55
      YDJ=ABS(HF)                                       ALTM   56
      CDJ=0.0                                           ALTM   57
      ER=FNORM(G)                                       ALTM   58
      ER1=1.0/ER                                        ALTM   59
      CALL VSCL(ER1,G,RMDP)                             ALTM   60
      YRJ=HR                                            ALTM   61
      CRJ=0.0                                           ALTM   62
      RMDL=-1.0                                         ALTM   63
      NOMES=1                                           ALTM   64
      RETURN                                            ALTM   65
C     END OF ALTM (SUBROUTINE)                          ALTM   66
```

```
C
C     START OF POSM (SUBROUTINE)
      ENTRY POSM
      RETURN
C     END OF POSM (SUBROUTINE)
C
C     START OF TDPM (SUBROUTINE)
      ENTRY TDPM
      IF (NB.EQ.0) GO TO 51
C     SETUP GROUND XMTTR MEASUREMENT
 51   CONTINUE
      IF (NA.EQ.0) RETURN
C     SETUP NAVSTAT
 52   CONTINUE
      K1=1
      DO 28 K2=2,10
      IF (TQ(K2).GT.FLTIME) GO TO 28
      TQ(K1)= TRSW(K2)+FLTIME
      GO TO (40,41,42,43,44,45,46,47),K1
 40   CONTINUE
      CALL TAAM
      GO TO 48
 41   CONTINUE
      CALL TEWM
      GO TO 48
 42   CONTINUE
      CALL TALM
      GO TO 48
 43   CONTINUE
      CALL TRRM
      GO TO 48
 44   CONTINUE
      CALL TPCM
      GO TO 48
 45   CONTINUE
      CALL TMOM
      GO TO 48
 46   CONTINUE
```

```
      CALL TMMM                                                          TOPM
      GO TO 48                                                           TOPM
 47   CONTINUE                                                           TOPM
      CALL TDSM                                                          TOPM
 48   K1=K1+1                                                            TOPM
 28   CONTINUE                                                           TOPM
      RETURN                                                             TOPM
C     END OF TOPM (SUBROUTINE)                                           TOPM
      END                                                                TOPM
      SUBROUTINE TRRM                                                    TRRM
C     (TO RUN PROGRAM,COMMON BLOCK DATA AT END OF THIS LISTING           TRRM
C     MUST BE INSERTED)                                                  TRRM
C                                                                        TRRM
C     START OF TRRM (SUBROUTINE)                                         TRRM
      ENTRY TRRM                                                         TRRM
C     VECTOR RANGE                                                       TRRM
      GO TO (60,61,62),IRT                                               TRRM
 60   CONTINUE                                                           TRRM
      GO TO 63                                                           TRRM
 61   CONTINUE                                                           TRRM
      GO TO 63                                                           TRRM
 62   CONTINUE                                                           TRRM
      GO TO 59                                                           TRRM
 63   CONTINUE                                                           TRRM
 58   CONTINUE                                                           TRRM
      CALL VADD(P,D,VEC1)                                                TRRM
      CALL VSUB(VEC1,EJ ,RJ)                                             TRRM
      YDJ=FNORM(RJ)                                                      TRRM
      UR=VNORM(RJ,UJ)                                                    TRRM
      GO TO (64,65,66),IRT                                               TRRM
 64   CONTINUE                                                           TRRM
      CALL VMOV(DUD,DD,3)                                                TRRM
      CALL VADD(DP,DD,VEC)                                               TRRM
      CALL VRXC(UJ,VEC,RJD)                                              TRRM
      RETURN                                                             TRRM
 59   CONTINUE                                                           TRRM
C     NAVSTAT DATA WORD                                                  TRRM
      GO TO 58                                                           TRRM
 65   CONTINUE                                                           TRRM
```

```
      CALL VSUB(DUD,DEMD,DD)                                            TRRM  31
      GO TO 67                                                          TRRM  32
   66 CONTINUE                                                          TRRM  33
      CALL VMOV(DUD,DD,3)                                               TRRM  34
   67 CONTINUE                                                          TRRM  35
      CALL VADD(DP,DD,VEC)                                              TRRM  36
      CALL VSUB(VEC,EJD,VEC1)                                           TRRM  37
      CALL VRXC(UJ,VEC1,RJD)                                            TRRM  38
      RETURN                                                            TRRM  39
C     END OF TRRM (SUBROUTINE)                                          TRRM  40
C                                                                       TMOM   1
C     START OF TMOM (SUBROUTINE)                                        TMOM   2
      ENTRY TMOM                                                        TMOM   3
      RMJ=(OK1*RMJ-DRMJ)-VLITE*(DTU-DTJ)                                TMOM   4
      RMJD=(OK2*RMJD-DRJD)-VLITE*(DTU-DTJ)                              TMOM   5
      YRJ= ER-RMJ                                                       TMOM   6
      YRRJ= ERD-RMJD                                                    TMOM   7
      RETURN                                                            TMOM   9
C     END OF TMOM (SUBROUTINE)                                          TMOM  10
C     START OF TMMM (SUBROUTINE)                                        TMMM   1
C                                                                       TMMM   2
      ENTRY TMMM                                                        TMMM   3
      CALL VMOV(UJ,RMDP,3)                                              TMMM   4
      CALL VSCL(-1.,UJ,RMDE)                                            TMMM   5
      RMDTU(1)=VLITE                                                    TMMM   6
      RMDTU(2)=0.                                                       TMMM   7
      RMDTJ(1)=-VLITE                                                   TMMM   8
      RMDL =1.0                                                         TMMM   9
      CALL VMOV(UJ,RRDPD,3)                                             TMMM  10
      RRTUD = VLITE                                                     TMMM  11
      RRTJD =-VLITE                                                     TMMM  12
      CALL MGID (VEC2,4)                                                TMMM  13
      CALL VCXR (UJ,UJ,VEC3)                                            TMMM  14
      CALL SUBM (VEC2,VEC3,VEC4,3,3)                                    TMMM  15
      DTN= 1.0/ER                                                       TMMM  16
      CALL MSCL (DTN,VEC4,VEC3)                                         TMMM  17
      GO TO (68,69,69),IRT                                              TMMM  18
   68 CONTINUE                                                          TMMM  19
```

```
      CALL VADD (DP,DD,VEC)                                   TMMM  20
      CALL HPYM (VEC,VEC3,RRDP,1,3,3)                         TMMM  21
      CALL VSCL (-1.,RRDP,RRDE)                               TMMM  22
      RETURN                                                  TMMM  23
 69   CALL VSUB (DP,DD,VEC)                                   TMMM  24
      CALL VSUB (VEC,EJD,VEC1)                                TMMM  25
      CALL MPYM (VEC1,VEC3,RRDP,1,3,3)                        TMMM  26
      CALL VSCL (-1.,RRDP,RRDE)                               TMMM  27
      RETURN                                                  TMMM  28
C     END OF TMMM (SUBROUTINE)                                TMMM  29
C     START OF TPCM (SUBROUTINE)                              TPCM   1
C                                                             TPCM   2
      ENTRY TPCM                                              TPCM   3
      DRJ = DPTJ+DPIJ+DLJ                                     TPCM   4
      RETURN                                                  TPCM   5
C     END OF TPCM (SUBROUTINE)                                TPCM   6
C                                                             TEWM   1
C     START OF TEWM (SUBROUTINE)                              TEWM   2
      ENTRY TEWM                                              TEWM   3
      TM=RO+HT(NC)                                            TEWM   4
      CALL MXMV2(VEC,TN,3,1,NC)                               TEWM   5
      CALL VSCL(TM,VEC,VEC)                                   TEWM   6
      CALL VSUB(VEC,DEJ,EJ)                                   TEWM   7
      RETURN                                                  TEWM   8
C     END OF TEWM (SUBROUTINE)                                TEWM   9
C     START OF TAAM (SUBROUTINE)                              TAAM   1
C                                                             TAAM   2
      ENTRY TAAM                                              TAAM   3
      RETURN                                                  TAAM   4
C     END OF TAAM (SUBROUTINE)                                TAAM   5
C                                                             TALM   1
C     START OF TALM (SUBROUTINE)                              TALM   2
      ENTRY TALM                                              TALM   3
      RETURN                                                  TALM   4
C     END OF TALM (SUBROUTINE)                                TALM   5
C     START OF TDSM (SUBROUTINE)                              TALM   6
C                                                             TALM   7
      ENTRY TDSM                                              TALM   8
```

```
      NOMES=IRT+1                                                       TALM   9
C     2=GROUND,3=AIRBORNE,4=NAVSTAT                                     TALM  10
      CDJ=0.                                                            TALM  11
      CRJ=0.                                                            TALM  12
      RETURN                                                            TALM  13
C     END OF TDSM (SUBROUTINE)                                          TALM  14
      END                                                               TALM  15
C     END OF SUBROUTINE REF-NAV                                         TALM  16
C                                                                       KLMN   1
C     KALMAN MODULE GROUPS                                              KLMN   2
C     START OF KALMAN SUBROUTINE                                        KLMN   3
C     (TO RUN PROGRAM,COMMON BLOCK DATA AT END OF THIS LISTING          KLMN   4
C     MUST BE INSERTED)                                                 KLMN   5
      NAMELIST/PKAL/P11,P13,P14,P31,P33,P34,P41,P43,P44,OD11,OD13       KLMN   6
     1,GD13,R,AD4,AD44,AURA,ARUC,AREJ                                   KLMN   7
     2,NOMES,DSWTH,RSWTH,NRSW,NOS1,NOMS,NOEM,NOSW,REQM,REQO             KLMN   8
     3,P55,P66,P77                                                      KLMN   9
      KFS=1                                                             KLMN  10
      KCRNT = 1                                                         KLMN  11
      KCOUNT = 1                                                        KLMN  12
      KALL = 1                                                          KLMN  13
      CALL VMOV(V,VO,3)                                                 KLMN  14
      DO 8 I=1,539                                                      KLMN  15
      VO(I)=0.0                                                         KLMN  16
      IF (I.GT.50) GO TO 8                                              KLMN  17
      RKARY(I)=1.0                                                      KLMN  18
 8    CONTINUE                                                          KLMN  19
      FTKAL=0.0                                                         KLMN  20
      READ (5,PKAL)                                                     KLMN  21
      WRITE (6,PKAL)                                                    KLMN  22
      RETURN                                                            KLMN  23
C                                                                       KLMN  24
      ENTRY KALMN                                                       KLMN  25
 10   CONTINUE                                                          KLMN  26
      IF (FLTIME.GE.FTKAL) GO TO 11                                     KLMN  27
      DO 12 I=2,10                                                      KLMN  28
      IN = I                                                            KLMN  29
      IF (FLTIME.GT.VK(I))   GO TO 12                                   KLMN  30
```

```
      VK(I) = TKLT(I) + FLTIME                                  KLMN
      IF (IN.GT.3) GO TO 13                                     KLMN
      GO TO (14,14,15,13),IN                                    KLMN
   14 CONTINUE                                                  KLMN
      CALL KSWM                                                 KLMN
      GO TO 12                                                  KLMN
   15 CONTINUE                                                  KLMN
      IF (DSWTH.EQ.2) RETURN                                    KLMN
      IF (NOMES.EQ.0) GO TO 13                                  KLMN
      CALL KMMM                                                 KLMN
      NOMES = 0                                                 KLMN
      GO TO 12                                                  KLMN
   13 CONTINUE                                                  KLMN
      IF (FLTIME.LT.TKAL) RETURN                                KLMN
      IF (KALL.EQ.2) RETURN                                     KLMN
      GO TO (17,18,19,20,21,22),KCOUNT                          KLMN
   17 CONTINUE                                                  KLMN
      CALL KMRM                                                 KLMN
      GO TO 23                                                  KLMN
   18 CONTINUE                                                  KLMN
      CALL KMCM                                                 KLMN
      GO TO 23                                                  KLMN
   19 CONTINUE                                                  KLMN
      CALL KMOM                                                 KLMN
      GO TO 23                                                  KLMN
   20 CONTINUE                                                  KLMN
      IF (KFS.EQ.2) CALL KFCW                                   KLMN
      KFS=1                                                     KLMN
      CALL KFIM                                                 KLMN
      GO TO 23                                                  KLMN
   21 CONTINUE                                                  KLMN
      IF (KFS.EQ.2) CALL KFCW                                   KLMN
      KFS = 1                                                   KLMN
      CALL KCOM                                                 KLMN
      KALL = 2                                                  KLMN
   23 KCOUNT = KCOUNT + 1                                       KLMN
   12 CONTINUE                                                  KLMN
      RETURN                                                    KLMN
   11 CONTINUE                                                  KLMN
```

```
      IF (KFFN.EQ.3) KFS=2
      KFFN=2
      FTKAL = FTKAL+ TKAL
      KCRNT=1
      IF (DSWTH.EQ.2) GO TO 22
   24 CONTINUE
      IF (RSWTH.EQ.2) GO TO 25
   26 CONTINUE
C     UPDATE PHI TO END OF  INTERVAL
      CALL KTMM
      CALL KTUM(1)
      IF (NOMS.EQ.0) RETURN
      DO 261 K = 1,NOMS
      NOS1 = K
      CALL KSYN
      CALL KMMM1
  261 CONTINUE
      NOS1=1
      WRITE(6,1025)
 1025 FORMAT(1H0,(9(2X,F10.5)))
 1021 FORMAT(2X,(9(2X,F10.7)))
      KALL = 1
      KCOUNT = 1
      KCRNT = 1
      NOMS = 0
      RETURN
   22 CONTINUE
      DO 27 K=1,NOSW
      NOS1 = K
      CALL KTMM
      CALL KTUM(1)
   27 CONTINUE
      KCOUNT=5
      DSWTH =1
      KCRNT= 2
      KALL = 1
      GO TO 24
   25 CONTINUE
      DO 28 K = 1,NRSW
```

```
      CALL KTMM
      CALL KTUM (1)
 28   CONTINUE
      KALL = 1
      KCOUNT=1
      RSWTH =1
      IF (KCRNT.EQ.1) GO TO 26
      KCRNT =1
      KCOUNT=5
      KALL= 1
      RETURN
      END
C     END OF SUBROUTINE KALMAN
C
C
C
C
C     START OF KALMAN SUBROUTINE GROUP
      SUBROUTINE KSWM
C     (TO RUN PROGRAM,COMMON BLOCK DATA AT END OF THIS LISTING
C     MUST BE INSERTED)
      DIMENSION SUM(6),SUM1(6),SUM2(6),BDRM(50),BDMS(50),NKK1(6)
C
C     START OF KSWM (SUBROUTINE)
      ENTRY KSWM
      IF (DEQM.NE.KEQM) GO TO 50
      IF (DRMM.NE.KEQM) GO TO 50
 49   CONTINUE
      IF (REQM.NE.REQO) GO TO 800
      RETURN
 50   CONTINUE
      IF (FTKAL.NE.O.) GO TO 66
 74   CONTINUE
      GO TO (60,61,62,62,62),KRMM
C     INITIALIZE FOR PDR MODE
 60   CONTINUE
      GO TO 65
C     INITIALIZE FOR ADR MODE
 61   CONTINUE
```

```
      GO TO 65                                                          KSWM
C     INITIALIZE FOR IDR MODE                                           KSWM
 62   CONTINUE                                                          KSWM
      GO TO 65                                                          KSWM
 65   CONTINUE                                                          KSWM
      KRMM = DRMM                                                       KSWM
      KRMO = DRMM                                                       KSWM
      KEQM = DEQM                                                       KSWM
      DSWTH = 2                                                         KSWM
      GO TO 49                                                          KSWM
C     INFLIGHT DR-NAV. SWITCHING                                        KSWM
 66   CONTINUE                                                          KSWM
      IF (DSWTH.EQ.2) GO TO 67                                          KSWM
      DSWTH = 2                                                         KSWM
      NOSW = 0                                                          KSWM
 67   NOSW = NOSW + 1                                                   KSWM
C     SAVE (O,G,GR FOR COMD,PRED+COMD AND OR,RR COMR MATRICES)          KSWM
C     MORE THAN (4) DRNAV SWITCH IN ONE KALMAN INTERVAL WILL RESULT     KSWM
C     IN A RETURN TO PDR MODE.,(DEQM=1=NO NAV.EQUIPMENT)                KSWM
      IF (NOSW.LT.4) GO TO 69                                           KSWM
      DEQM = 1                                                          KSWM
      GO TO 60                                                          KSWM
 69   CONTINUE                                                          KSWM
      IKK(NOSW) = KRMO                                                  KSWM
      GO TO (70,71,72,72,72),KRMO                                       KSWM
C     SAVE IDR (O,G,R)                                                  KSWM
 72   CONTINUE                                                          KSWM
      KC = NOSW                                                         KSWM
      CT = FTKAL-FLTIME                                                 KSWM
C     MOVE THE PHI SUBMATRIX                                            KSWM
      CALL MXMV1(OD11,PHI,36,1,KC)                                      KSWM
      CALL MXMV1(AD44,PHI,9, 2,KC)                                      KSWM
      CALL MXMV1(CT  ,PHI,1, 3,KC)                                      KSWM
      CALL MXMV1(DV  ,PHI,3, 4,KC)                                      KSWM
      CALL MXMV1(DP  ,PHI,3, 5,KC)                                      KSWM
      CALL MXMV1(WPC ,PHI,3, 6,KC)                                      KSWM
      CALL MXMV1(TPC ,PHI,9, 7,KC)                                      KSWM
      GO TO 73                                                          KSWM
 71   CONTINUE                                                          KSWM
```

```
      GO TO 73
C     SAVE POR (O,G,R)
 70   CONTINUE
      GO TO 73
C     TURN OF KTAM,PROCESSING;TO BY PASS KFIM'S PROCESSING NEXT INT.
 73   CONTINUE
      GO TO 74
 800  CONTINUE
      IF (FTKAL.NE.0.0) GO TO 801
C     INITIALIZE REF.NAV VARIABLES
      GO TO 802
 801  IF (RSWTH.EQ.2) GO TO 80
      RSWTH = 2
      NRSW= 0
 80   NRSW = NRSW+1
 802  CONTINUE
      DO 81 I=1,50,5
      IF (RKARY(I).EQ.1) GO TO 81
      IF (RKARY(I).GT.3) GO TO 88
      IN = I
      IM = IN + 4
      IMM = IN + 1
      DO 82 I1 = IMM,IM
      I2=RKARY(I1)
      GO TO (82,84,85),I2
C     ADD NEW EMITTER TO NET;ADD (OR AND RR FOR THIS EMITTER)
 84   CONTINUE
      GO TO 86
C     DROP THIS EMITTER FROM NET; DROP (OR AND RR FOR THIS EMITTER)
C     SAVE (PRED+COMDR OR AND RR)
 85   CONTINUE
 86   RKARY(I1)=1.0
 82   CONTINUE
      GO TO 81
 88   IF (RKARY(I).GT.4) GO TO 89
C     ADD A COMPLETE NET; ENCLUDING (OR AND RR)
      GO TO 90
C     DELETE THE ENTIRE NET:SAVE (PRED+COMP OR AND RR)
```

```
 89   CONTINUE
 90   RKARY(I)=1.0
 81   CONTINUE
      REQO =REQM
      RETURN
C     END OF KSWM (SUBROUTINE)
C
C     START OF KTMM (SUBROUTINE)
      ENTRY KTMM
      V1=FNORM(VO)
      CALL VMOV(V,VO,3)
      ISEQ = 1
      IF (DSWTH.NE.2) GO TO 10
      ISEQ = 2
      GO TO 10
      ENTRY KSYN
      V1 = FNORM(V)
      ISEQ = 3
 10   CONTINUE
      IF (ISEQ.GE.2) GO TO  25
C     COMPUTE   O44(TF -TS)  END OF INTERVAL
      CT = TFKAL-FLTIME
 26   CALL VSCL (CT,AD4,VEC1)
      CALL MGID (OD44,4)
      CALL MPVC (OD44,VEC1,4)
      GO TO 27
C     COMPUTE O44 (TF-TI) MEAS.SYNC OR DR-SWITCH
 25   CONTINUE
      CT = PHI(1,3,NOS1)
      GO TO 26
C     COMPUTE O34;G33
 27   CONTINUE
      CALL MGID (VEC2,4)
      CALL MSCL(CT,VEC2,OD34)
      CALL MXMV (OD34,GD33,9)
      IF (ISEQ.GE.2) GO TO 28
C     COMPUTE O33 (TF-TS) END OF INTERVAL
      CALL MTRN (TPC,WEI,VEC3)
      CALL VSCL (CT,VEC3,VEC4)
```

```
      CALL VSCL (CT,WPC,VEC5)                                  KTMM   34
 29   CALL VADD (VEC5,VEC4,VEC1)                               KTMM   35
      CALL WCROS (VEC1,VEC4)                                   KTMM   36
      CALL MGID (OD33,4)                                       KTMM   37
      CALL SUBM(OD33,VEC4,OD33,3,3)                            KTMM   38
      GO TO 30                                                 KTMM   39
C     COMPUTE O33 (TF -TI) MEAS.SYN OR DR-SWITCH               KTMM   40
 28   CONTINUE                                                 KTMM   41
      CALL MXMV2 (VEC1,PHI,3,6,NOS1)                           KTMM   42
      CALL MXMV2 (VEC2,PHI,9,7,NOS1)                           KTMM   43
      CALL MTRN (VEC2,WEI,VEC3)                                KTMM   44
      CALL VSCL (CT,VEC1,VEC5)                                 KTMM   45
      CALL VSCL (CT,VEC3,VEC4)                                 KTMM   46
      GO TO 29                                                 KTMM   47
 30   DO 301 I=1,3                                             KTMM   48
      SUM(I) =G(I)                                             KTMM   49
      SUM1(I)=PHI(I,4,NOS1)                                    KTMM   50
      SUM2(I)=PHI(I,5,NOS1)                                    KTMM   51
 301  CONTINUE                                                 KTMM   52
      IF (ISEQ.EQ.1) GO TO 31                                  KTMM   53
C     COMPUTE  OD13,GD13 (TF - TI) MEAS.SYNC.& ORSWITCH        KTMM   54
      CALL PO1 (SUM,SUM1,SUM2,CT,V1)                           KTMM   55
 32   CALL MXMV(TPC,VEC2,9)                                    KTMM   56
      CALL INVERT(VEC2,3,3,DET)                                KTMM   57
      CALL WCROS(SUM,VEC3)                                     KTMM   58
      CALL WCROS(SUM1,VEC4)                                    KTMM   59
      CALL WCROS(SUM2,VEC5)                                    KTMM   60
      CALL MTRN (VEC4,VEC2,OD131)                              KTMM   61
      CALL MTRN (VEC3,VEC2,OD132)                              KTMM   62
      CALL MTRN (VEC5,VEC2,GD131)                              KTMM   63
      CA = CT*CT*.5                                            KTMM   64
      CALL VSCL(CA,G,SUM)                                      KTMM   65
      CALL WCROS(SUM,VEC3,VEC2)                                KTMM   66
      CALL MTRN(VEC4,VEC2,GD132)                               KTMM   67
      GO TO 33                                                 KTMM   68
C     COMPUTE OD13 GD13 (TF-TS) END OF INTERVAL                KTMM   69
 31   CALL PO1 (SUM,SUM1,SUM2,CT,V1)                           KTMM   70
      GO TO 32                                                 KTMM   71
C     COMPUTE ADC11 AND OD11                                   KTMM   72
```

```
33    CONTINUE
      IF (ISEQ.GE.2) GO TO 34
      RQ=VNORM(G,VEC1)
      GO TO 35
34    CONTINUE
      RQ=VNORM(G,VEC1)
35    CONTINUE
      CALL VCXR(VEC1,VEC1,VEC2,3,3)
      CALL MSCL(3.0,VEC2,VEC3)
      CALL MGID(VEC2,4)
      CALL SUBM(VEC2,VEC3,VEC4,3,3)
      CALL MSCL(-RTS,VEC4,TO21)
      CALL VSCL(-2.0,WEI,VEC1)
      CALL WCROS(VEC1,TO22)
      CALL MGID(TO12,4)
      CALL MSCL(0.0,VEC5,TO11)
      CALL MXMV(OD11,VEC12,36)
      CALL ARRY(OD11,VEC12,1)
      CALL VSCL1(CT,OD11,VEC12,36,1)
      CALL MPYM(VEC12,VEC12,VEC13,6,6,6)
      CALL VSCL1(0.5,VEC13,VEC13,36,1)
      CALL ADDM(VEC12,VEC13,6,6)
      CALL MGID(VEC14,7)
      CALL ADDM(VEC14,VEC13,OD11,6,6)
      OURA = 1. + ARUA*CT
      CALL MGID (VEC2,4)
      CALL MXMV (VEC2,OURC,4)
      OURC(1) = CT
      OURC(3) = CT
      OURC(4) = ARUC*CT
      CALL MGID (ORE,5)
      CALL MXMV(OD13,VEC12,18)
      CALL ARRY(OD13,VEC12,2)
      CALL MXMV(GD13,VEC12,18)
      CALL ARRY(GD13,VEC12,2)
      ORUA=1.+ARUA*CT
      CALL VSCL1(CT,ARUC,VEC1,4,1)
      CALL MGID(VEC1,3)
      CALL ADDM(VEC1,VEC4,OURC,2,2)
```

```
      CALL MGID(DURE,4)
      NS=PHI(1,13,NOS1)
      DREJ(NS)=1.+AREJ(NS)*CT
      RETURN
C
C     START OF KMMM (SUBROUTINE)
      ENTRY KMMM
C     SAVE MEASUREMENT DATA TO END OF KALMAN INTERVAL
      CT = FTKAL - FLTIME
      CALL VSCL(0.0,VEC1,VEC1)
      CALL VMOV(RMOP,VEC1,3)
C     SAVE DATA IN PHI ARRAY
  126 CONTINUE
      CALL MXMV(TPC,VEC3,9)
      DO 127 I=1,9
      PHI(I,7,NOS1)=VEC3(I)
      IF (I.GT.6) GO TO 128
      PHI(I,1,NOS1) = VEC1(I)
  128 IF (I.GT.3) GO TO 127
      PHI(I,4,NOS1) = DV(I)
      PHI(I,5,NOS1) = DP(I)
      PHI(I,6,NOS1) = WPC(I)
      PHI(I,11,NOS1)=RMDE(I)
  127 CONTINUE
      PHI(1,14,NOS1)=RMOTU(1)
      PHI(2,14,NOS1) = 0.0
      PHI(1,2,NOS1)=ER
      PHI(1,3,NOS1) = CT
      PHI(1,8,NOS1) = CDJ
      PHI(1,9,NOS1) = YRJ
      PHI(1,10,NOS1) = CRJ
      PHI(1,15,NOS1)=RMOL
      PHI(1,12,NOS1) = NOMES
      PHI(1,13,NOS1)=NC
      NOMS = NOMS+1
      NOS1=NOS1+1
      RETURN
      ENTRY KMMM1
      IF (NOS1.NE.1) GO TO 129
```

```
      DO 130 K = 1,4                                    KMMM
      DO 130 J = 1,6                                    KMMM
      DO 130 I = 1,50                                   KMMM
      DDMS(I,J,K) = 0.0                                 KMMM
  130 CONTINUE                                          KMMM
  129 CONTINUE                                          KMMM
      GO TO (131,132,133,133,133,133),DRMM              KMMM
C     PDR MODE                                          KMMM
  131 CONTINUE                                          KMMM
      RETURN                                            KMMM
C     ADR MODE                                          KMMM
  132 CONTINUE                                          KMMM
      RETURN                                            KMMM
C     IDR MODE                                          KMMM
  133 CONTINUE                                          KMMM
C     GET (MDJ,MRJ)                                     KMMM
      CALL MXMV2(VEC2,PHI,1,6,NOS1)                     KMMM
      CALL MXMV2(VEC ,PHI,11,3,NOS1)                    KMMM
      CALL MPYM (VEC2,OD11,VEC4,1,6,6)                  KMMM
      CALL MPYM (VEC2,GD13,VEC5,1,6,3)                  KMMM
      CALL MPYM (VEC2,OD13,VEC3,1,6,3)                  KMMM
      CALL MXMV3(VEC4,DMS,7,2,1)                        KMMM
      CALL MXMV3 (VEC3,DMS1,3,2,1)                      KMMM
      CALL MXMV3 (VEC5,DMS1,3,2,2)                      KMMM
      CALL MXMV4 (VEC,DDMS,1,NT,NOS1,PHI)               KMMM
      NT = PHI(1,12,NOS1)                               KMMM
      GC TO (134,135,136,137,138),NT                    KMMM
C     ALTITUDE  MEASUREMENT SUMATION                    KMMM
  134 CONTINUE                                          KMMM
      DMR(1,1,1)=(-1.*OURA)+DMR(1,1,1)                  KMMM
      GO TO 1341                                        KMMM
C     GROUND EMITTER                                    KMMM
  135 CONTINUE                                          KMMM
      NS=PHI(1,13,NOS1)                                 KMMM
      CALL VSCL(0.,VEC,VEC)                             KMMM
      CALL MXMV2(VEC,PHI,14,2,NOS1)                     KMMM
      CALL MXMV2(VEC1,PHI,11,3,NOS1)                    KMMM
      CALL MXMV2(VEC3,PHI,15,1,NOS1)                    KMMM
      CALL MPYM(VEC,OURC,VEC3,1,2,2)                    KMMM
```

```
      CALL MXMV6(VE:3,DMR,2,1,1,PHI,2)
      CALL MPYM(VEC1,OURE,VEC4,1,3,3)
      CALL MXMV6(VLC4,DMR,3,1,1,PHI,4)
      GO TO 1341
C     LOS
 136  CONTINUE
      RETURN
C     NAVSTAT
 137  CONTINUE
      NS=PHI(1,13,NOS1)
      CALL MXMV2(VEC,PHI,14,2,NOS1)
      CALL MXMV2(VEC1,PHI,11,3,NOS1)
      CALL MXMV2(VEC3,PHI,15,1,NOS1)
      CALL MPYM(VEC,OURA,VEC3,1,2,2)
      CALL MXMV6(VEC3,DMR,2,1,1,PHI,2)
      CALL VSCL(0.0,VEC1,VEC1,6,1)
      VEC1(NS)=OREJ(NS)
      CALL MXMV6(VEC1,DMR,4,1,1,PHI,4)
      GO TO 1341
C     LORAN
 138  CONTINUE
 1341 CONTINUE
      IF (NOS1.NE.NOMS) RETURN
      CALL MXMV5 (VEC,DDMS,1,1,NOS1,PHI)
      RETURN
C     END OF KMMM (SUBROUTINE)
C
C     START OF KCOM (SUBROUTINE)
      ENTRY KCOM
      CALL KTUM (2)
      CALL VSUB1(XDS,UDS,XDS,12,1)
      CALL VTRN (TLC,XDS,VEC)
      CALL VTRN (TKL,VEC,VEC1)
      CALL VTRN (TPC,VEC1,VEC)
      CALL VMOV(XDS(7),VEC3,3)
      CALL VSUB(VEC,VEC3,VEC1)
      T1 = 1.0/TKAL
      CALL VSCL (T1,VEC1,WL)
```

```
      CALL VSUB(P,UDS,P)                                              KCOM  13
      CALL VMOV(UDS(4),VEC,3)                                         KCOM  14
      CALL VSUB(V,VEC,V)                                              KCOM  15
      RETURN                                                          KCOM  16
      END                                                             KCOM  17
C     END OF KCOM (SUBROUTINE)                                        KCOM  18
C     MUST BE INSERTED)                                               KTUM   1
C     (TO RUN PROGRAM,COMMON BLOCK DATA AT END OF THIS LISTING        KTUM   2
C                                                                     KTUM   3
C     START OF KTUM (SUBROUTINE)                                      KTUM   4
      SUBROUTINE KTUM(IX)                                             KTUM   5
      IF(IX.LE.0.OR.IX.GT.2) IX=1                                     KTUM   6
C     X = PHI*X+G                                                     KTUM   7
      CALL GET(XDS,VEC6,VEC1,VEC,VEC5)                                KTUM   8
      CALL MSCL(0.0,VEC4,VEC4)                                        KTUM   9
      CALL MPYM(OD11,VEC6,VEC3,6,6,1)                                 KTUM  10
      CALL MPYM(OD13,VEC1,VEC4,6,3,1)                                 KTUM  11
      CALL VADD1(VEC3,VEC4,XDS,6,1)                                   KTUM  12
      CALL MPYM(OD33,VEC1,VEC3,3,3,1)                                 KTUM  13
      CALL MPYM(OD34,VEC,VEC4,3,3,1)                                  KTUM  14
      CALL VADD1(VEC3,VEC4,XDS,3,7)                                   KTUM  15
      CALL MPYM(OD44,VEC,VEC3,3,3,1)                                  KTUM  16
      CALL VSCL(0.0,VEC,VEC)                                          KTUM  17
      CALL VADD1(VEC3,VEC,XDS,3,11)                                   KTUM  18
C     DX + GU                                                         KTUM  19
      CALL MPYM(GD13,VEC5,VEC3,6,3,1)                                 KTUM  20
      CALL VADD1(VEC3,XDS,XDS,6,1)                                    KTUM  21
      CALL MPYM(GD33,VEC5,VEC3,3,3,1)                                 KTUM  22
      CALL VADD1(VEC3,XDS,XDS,3,7)                                    KTUM  23
      IF (IX.EQ.2) RETURN                                             KTUM  24
C     COMPUTE POT                                                     KTUM  25
      CALL MTRA(OD11,VEC12,6,6)                                       KTUM  26
      CALL MPYM(P11,VEC12,VEC13,6,6,6)                                KTUM  27
      CALL MXMV(VEC13,P11,36)                                         KTUM  28
      CALL MPYM(P31,VEC12,VEC13,3,6,6)                                KTUM  29
      CALL MXMV(VEC13,P31,18)                                         KTUM  30
      CALL MPYM(P41,VEC12,VEC13,3,6,3)                                KTUM  31
      CALL MXMV(VEC13,P41,18)                                         KTUM  32
      CALL MTRA(OD13,VEC12,6,3)                                       KTUM  33
```

```
      CALL MPYM(P13,VEC12,VEC13,6,3,6)                        KTUM  34
      CALL MADD(P11,VEC13,P11,6,6)                            KTUM  35
      CALL MPYM(P33,VEC12,VEC13,3,3,6)                        KTUM  36
      CALL ADDM (P31,VEC13,P31,3,3)                           KTUM  37
      CALL MPYM(P43,VEC12,VEC13,3,3,6)                        KTUM  38
      CALL MADD(P41,VEC13,P41,3,6)                            KTUM  39
      CALL MTRA(OD33,VEC12,3,3)                               KTUM  40
      CALL MPYM(P13,VEC12,VEC13,6,3,3)                        KTUM  41
      CALL MXMV(VEC13,P13,18)                                 KTUM  42
      CALL MTRT(P33,OD33,VEC13)                               KTUM  43
      CALL MXMV(VEC13,P33,9)                                  KTUM  44
      CALL MTRT(P43,OD33,VEC13)                               KTUM  45
      CALL MXMV(VEC13,P43,9)                                  KTUM  46
      CALL MPYM(P14,VEC12,VEC13,6,3,3)                        KTUM  47
      CALL MADD(P13,VEC13,P13,18)                             KTUM  48
      CALL MTRT(P34,OD34,VEC13)                               KTUM  49
      CALL MADD(P33,VEC13,3,3)                                KTUM  50
      CALL MTRT(P44,OD34,VEC13)                               KTUM  51
      CALL MADD(P43,VEC13,3,3)                                KTUM  52
      CALL MTRA(OD44,VEC12,3,3)                               KTUM  53
      CALL MPYM(P14,VEC12,VEC13,6,3,6)                        KTUM  54
      CALL MXMV(VEC13,P14,18)                                 KTUM  55
      CALL MTRT(P34,OD44,VEC13)                               KTUM  56
      CALL MXMV(VEC13,P34,9)                                  KTUM  57
      CALL MTRT(P44,OD44,VEC13)                               KTUM  58
      CALL MXMV(VEC13,P44,9)                                  KTUM  59
C     OPOT + R                                                KTUM  60
C     P11                                                     KTUM  61
      CALL MPYM(OD11,P11,VEC12,6,6,6)                         KTUM  62
      CALL MPYM(OD13,P31,VEC13,6,3,3)                         KTUM  63
      CALL MXMV(VEC12,P11,36)                                 KTUM  64
      CALL MADD(P11,VEC13,P11,6,6)                            KTUM  65
C     P23                                                     KTUM  66
      CALL MPYM(OD11,P13,VEC12,6,6,3)                         KTUM  67
      CALL MPYM(OD13,P33,VEC13,6,3,3)                         KTUM  68
      CALL MXMV(VEC12,P13,18)                                 KTUM  69
      CALL MADD(P13,VEC13,P13,6,3)                            KTUM  70
      CALL MTRA(P13,P31,6,3)                                  KTUM  71
C     P14                                                     KTUM  72
```

```
      CALL MPYM(OD11,P14,VEC12,6,6,3)                                  KTUM   73
      CALL MPYM(OD13,P34,VEC13,6,3,3)                                  KTUM   74
      CALL MXMV(VEC12,P14,18)                                          KTUM   75
      CALL MADD(P14,VEC13,P14,6,3)                                     KTUM   76
      CALL MTRA(P14,P41,6,3)                                           KTUM   77
C     P33                                                              KTUM   78
      CALL MTRN(OD33,P33,VEC12)                                        KTUM   79
      CALL MTRN(OD34,P43,VEC13)                                        KTUM   80
      CALL MADD(VEC12,VEC13,P33)                                       KTUM   81
C     P34                                                              KTUM   82
      CALL MTRN(OD33,P34,VEC12)                                        KTUM   83
      CALL MTRN(OD34,P44,VEC13)                                        KTUM   84
      CALL MADD(VEC12,VEC13,P34)                                       KTUM   85
      CALL MTRA(P34,P43,3,3)                                           KTUM   86
C     P44                                                              KTUM   87
      CALL MTRN(OD44,P44,VEC12)                                        KTUM   88
      CALL MXMV(VEC12,P44,9)                                           KTUM   89
      ORSA=ORUA*ORUA                                                   KTUM   90
C     PRSS P                                                           KTUM   91
      CALL VSCL1(ORSA,P55,P55,1)                                       KTUM   92
      CALL MTRA (OURC,VEC1,2,2)                                        KTUM   93
      CALL MPYM (P66,VEC1,VEC2,2,2,2)                                  KTUM   94
      CALL MPYM (OURC,VEC2,P66,2,2,2)                                  KTUM   95
      IF (IRT.EQ.2) GO TO 105                                          KTUM   96
      CALL MTRA (OURE,VEC2,3,3)                                        KTUM   97
      CALL MPYM (P77,VEC2,VEC3,3,3,3)                                  KTUM   98
      CALL MPYM (OURE,VEC3,P77,3,3,3)                                  KTUM   99
      GO TO 106                                                        KTUM  100
  105 CONTINUE                                                         KTUM  101
      CALL MTRA (OREJ,VEC12,4,4)                                       KTUM  102
      CALL MPYM (P77,VEC12,VEC13,4,4,4)                                KTUM  103
      CALL MPYM (OREJ,VEC13,P77,4,4,4)                                 KTUM  104
  106 CONTINUE                                                         KTUM  105
      CALL MPYM(OD11,P15,VEC12,6,6,1)                                  KTUM  107
      CALL MPYM(OD13,P35,VEC13,6,3,1)                                  KTUM  108
      CALL ADDM(VEC12,VEC13,P15,6,1)                                   KTUM  109
      CALL MPYM(OD11,P16,VEC12,6,6,2)                                  KTUM  110
      CALL MPYM(OD13,P36,VEC13,6,3,2)                                  KTUM  111
      CALL ADDM(VEC12,VEC13,P16,6,2)                                   KTUM  112
```

```
      CALL VSCL1(OURA,P15,6)
      CALL MTRA (P15,P51,6,1)
      CALL MPYM(OD33,P35,VEC12,3,3,1)
      CALL VSCL(OURA,VEC12,P35)
      CALL MTRA(P35,P53,3,1)
      CALL MPYM(OD33,P36,VEC12,3,3,2)
      CALL MPYM(OD44,P46,VEC13,3,3,2)
      CALL MTRA(ORUC,VEC1,2,2)
      CALL MPYM(VEC12,VEC1,P36,3,2,2)
      CALL MTRA(P36,P63,3,2)
      CALL MPYM(VEC13,VEC1,P46,3,2,2)
      CALL MTRA(P46,P64,3,2)
      CALL MTRA(P36,P63,3,2)
      CALL MPYM(OD44,P45,VEC2,3,3,1)
      CALL VSCL(OURC,VEC2,P45)
      CALL VMOV(P45,P54,3)
      CALL MPYM(P16,VEC1,P16,6,2,2)
      CALL MTRA(P16,P61,6,2)
      IF (IRT.GT.1) GO TO 107
      CALL MPYM(OD11,P17,VEC12,6,6,3)
      CALL MPYM(OD13,P37,VEC13,6,6,3)
      CALL ADDM(VFC12,VEC13,VEC14,6,3)
      CALL MPYM(OD33,P37,VEC12,3,3,3)
      CALL MPYM(OD44,P47,VEC13,3,3,3)
      CALL MTRA(OURE,VEC2,3,3)
      CALL MPYM(VEC14,VEC2,P17,6,3,3)
      CALL MTRA(P17,P71,6,3)
      CALL MPYM(VEC12,VEC2,P37,3,3,3)
      CALL MTRA(P37,P73,3,3)
      CALL MPYM(VEC13,VEC2,P47,3,3,3)
      CALL MTRA(P47,P74,3,3)
      GO TO 109
  107 CONTINUE
      CALL MPYM(OD11,P17,VEC12,6,6,4)
      CALL MPYM(OD13,P37,VEC13,6,6,4)
      CALL ADDM(VEC12,VEC13,VEC14,6,4)
      CALL MPYM(OD33,P37,VEC12,3,3,4)
      CALL MTRA(OREJ,VEC13,4,4)
```

```
      CALL MPYM(VEC14,VEC13,P17,6,4,4)
      CALL MTRA(P17,P71,6,4)
      CALL MPYM(OD44,P47,VEC14,3,3,4)
      CALL MPYM(VEC14,VEC13,P47,3,4,4)
      CALL MPYM(VEC12,VEC13,P37,3,4,4)
      CALL MTRA(P47,P74,3,4)
      CALL MTRA(P37,P73,3,4)
      CALL MTRA(OURC,VEC1,2,2)
      CALL MPYM(P56,VEC1,VEC3,1,2,2)
      CALL VSCL1(ORUA,VEC3,P56,2)
      CALL VMOV(P56,P65,2)
      IF (IRT.GT.1) GO TO 108
  109 CONTINUE
      CALL MTRA(OURE,VEC3,3,3)
      CALL MPYM(P57,VEC3,VEC4,1,3,3)
      CALL VSCL(ORUA,VEC4,P57)
      CALL VMOV(P57,P75,3)
      CALL MPYM(P67,VEC3,VEC4,2,3,3)
      CALL MPYM(OURC,VEC4,P67,2,2,3)
      CALL MTRA(P67,P76,2,3)
  108 CONTINUE
      CALL MTRA(OURE,VEC12,4,4)
      CALL MPYM(P57,VEC12,VEC13,1,4,4)
      CALL VSCL1(ORUA,VEC13,P57,4,1)
      CALL VMOV (P57,P75,4)
      CALL MPYM(P67,VEC12,VEC13,2,4,4)
      CALL MPYM(OURC,VEC13,P67,2,2,4)
      CALL MTRA(P67,P76,2,4)
C     ADD R
      CALL MPVC(P11,R,M,1)
      CALL MPVC(P33,R,4,7)
      CALL MPVC(P44,R,4,10)
      CALL MPVC(P55,R,2,13)
      CALL MPVC(P66,R,3,14)
      CALL MPVC(P77,R,5,16)
      RETURN
C     END OF KTUM (SUBROUTINE)
```

```
C
C       START OF KFIM (SUBROUTINE)
        ENTRY KFIM
        DO 300 J =1,6
        I1 = 1
        IF (DDMS(1,J,1).EQ.0.0) GO TO 300
C       GET, YD,YR,CD,CR AND DELTA Y
        DO 3011 I=2,6
        DMZ(I1)= DDMS(I,J,1)
 3011   I1=I1+1
C       GET THE MEASUREMENT MATRIX (M,N,& Z)
        K= 1
        DO 302 I=1,50
        BBDM(I)= 0.0
        DBDM(I)= 0.0
        BKDM(I)= 0.0
  302   CONTINUE
        K1=1
        DO 304 I=7,50
        DNS(K) = DDMS(I,J,2)
        HM(K) = DDMS(I,J,1)
        DZS(K)= DDMS(I,J,3)
  304   K=K+1
C       COMPUTE (MT AND YDM)
        I=1
        ETA=0.0
        QRM=0.
        URM=0.
        QDRM=0.
        CALL GET (HM,VEC6,VEC1,VEC,VEC5)
C        X11'=VEC6,X1'3=VEC1,X1'4=VEC,U1'3=VEC5
        CALL VSCL1(0.0,VEC12,VEC12,36,0)
        CALL VSCL1(0.0,VEC13,VEC13,36,0)
        CALL MPYM(P11,VEC6,VEC12,6,6,1)
        CALL MPYM(P13,VEC1,VEC13,6,3,1)
        CALL VADD1(VEC12,VEC13,VEC12,6,1)
        CALL MPYM(P14,VEC,VEC13,6,3,1)
        CALL VADD1(VEC12,VEC13,VEC12,6,1)
        CALL MPYM(P31,VEC6,VEC13,6,3,1)
```

```
      CALL VADD1(VEC12,VEC13,VEC12,3,7)                                 KFIM  40
      CALL MPYM(P33,VEC1,VEC11,3,3,1)                                   KFIM  41
      CALL VADD1(VEC12,VEC11,VEC12,3,7)                                 KFIM  42
      CALL MPYM(P34,VEC,VEC13,3,3,1)                                    KFIM  43
      CALL VADD1(VEC1,VEC13,VEC12,3,7)                                  KFIM  44
      CALL MPYM(P41,VEC6,VEC13,6,3,1)                                   KFIM  45
      CALL VADD1(VEC12,VEC13,VEC12,3,10)                                KFIM  46
      CALL MPYM(P43,VEC1,VEC13,3,3,1)                                   KFIM  47
      CALL MPYM(P44,VEC,VEC11,3,3,1)                                    KFIM  48
      CALL VADD1(VEC12,VEC11,VEC12,3,10)                                KFIM  49
      CALL MPYM(HM,VEC12,VEC13,1,12,1)                                  KFIM  50
      QDM = VEC13(1)                                                    KFIM  51
      CALL MPYM(HM,DZS,VEC13,1,12,1)                                    KFIM  52
      QDM = WDS-(2.0*VEC13(1))+ DMZ(3)+QDM                              KFIM  53
      CALL MPYM(HM,HM,VEC13,1,12,1)                                     KFIM  54
      UDM = VEC13(1)                                                    KFIM  55
      CALL VSUB1(VEC12,DZS,BBDM,12,1)                                   KFIM  56
      CALL GET(RM,VEC,VEC1,VEC2,VEC3)                                   KFIM  57
C     ALT=VEC(1),CLOCK=VEC1(1,2),EMITTER=VEC2(1,2,--4) MEARSUREMENTS    KFIM  58
      CLALB=VEC(1)*VEC(1)                                               KFIM  59
      CALL VSCL1(0.,VEC12,VEC12,36,1)                                   KFIM  60
      CALL VSCL1(VEC13,VEC13,36,1)                                      KFIM  61
      CALL MPYM(P66,VEC1,VEC4,2,2,1)                                    KFIM  62
      CALL VADD1(VEC12,VEC4,VEC12,2,2)                                  KFIM  63
      CALL VSCL1(CLALB,P55,VEC12,1,1)                                   KFIM  64
      CALL MPYM(P77,VEC2,VEC4,4,4,1)                                    KFIM  65
      CALL VADD1(VEC12,VEC4,VEC12,4,3)                                  KFIM  66
      CALL MPYM(RM,VEC12,VEC13,1,7,1)                                   KFIM  67
      QRM=VEC13(1)                                                      KFIM  68
      CALL MPYM(RM,DRZ,VEC13,1,7,1)                                     KFIM  69
      QRM=WRM-(2.0*VEC13(1)+DMZ(4)+QRM)                                 KFIM  70
      CALL VSUB1(VEC12,DRZ,BBRM,12,1)                                   KFIM  71
      CALL VSCL1(0.,VEC12,VEC12,36,1)                                   KFIM  72
      CALL VSCL1(0.,VEC13,VEC13,36,1)                                   KFIM  73
      CALL VSCL1(CLALB,P15,VEC12,6,1)                                   KFIM  74
      CALL MPYM(P16,VEC1,VEC4,6,2,1)                                    KFIM  75
      CALL VADD1(VEC12,VEC4,VEC12,6,1)                                  KFIM  76
      CALL MPYM(P17,VEC2,VEC4,6,4,1)                                    KFIM  77
      CALL VADD1(VEC12,VEC4,VEC12,6,1)                                  KFIM  78
```

```
      CALL  VSCL1(CLALB,P35,VEC12,3,7)                               KFIM
      CALL  MPYM(P36,VEC1,VEC4,3,2,1)                                KFIM
      CALL  VADD1(VEC12,VEC4,VEC12,3,7)                              KFIM
      CALL  MPYM(P37,VEC2,VEC4,3,4,1)                                KFIM
      CALL  VADD1(VEC12,VEC4,VEC12,3,7)                              KFIM
      CALL  VSCL1(CLALB,P45,VEC12,3,10)                              KFIM
      CALL  MPYM(P46,VEC1,VEC4,3,2,1)                                KFIM
      CALL  VADD1(VEC12,VEC4,VEC12,3,10)                             KFIM
      CALL  MPYM(P47,VEC2,VEC4,3,4,1)                                KFIM
      CALL  ADD1(VEC12,VEC4,VEC12,3,10)                              KFIM
      CALL  MPYM(P56,VEC1,VEC4,1,2,1)                                KFIM
      CALL  VADD1(VEC12,VEC4,VEC12,1,13)                             KFIM
      CALL  MPYM(P57,VEC2,VEC4,1,4,1)                                KFIM
      CALL  VADD1(VEC12,VEC4,VEC12,1,13)                             KFIM
      CALL  MPYM(P67,VEC2,VEC4,2,4,1)                                KFIM
      CALL  VADD1(VEC12,VEC4,VEC12,2,14)                             KFIM
      CALL  MPYM(HM,VEC12,VEC13,1,12,1)                              KFIM
      QDRM=VEC13(1)*2.                                               KFIM
      CALL  MPYM(RM,RM,VEC13,1,7,1)                                  KFIM
      URM=VEC13(1)                                                   KFIM
      CALL  VADD1(XRS,URS,VEC13,7,1)                                 KFIM
      CALL  MPYM(RM,VEC13,VEC14,1,7,1)                               KFIM
      YRMP=VEC14(1)+DMZ(2)                                           KFIM
      CALL  VMUV1(VEC12,BDRM,12,1)                                   KFIM
      CALL  VADD1(XDS,UDS,VEC13,12,1)                                KFIM
      CALL  MPYM(HM,VEC13,VEC12,1,12,1)                              KFIM
      YDMP=VEC12(1)                                                  KFIM
      CALL  MPYM (DNS,UDDS,VEC12,1,12,1)                             KFIM
      YDMP = (VEC12(1)-YDMP)+DMZ(1)                                  KFIM
      DYMP = YDMP-YRMP                                               KFIM
      QB   = QDM+QRM+2.0*QDRM                                        KFIM
      UM   = UDM+URM                                                 KFIM
      QB1= 1.0/QB                                                    KFIM
      UD1= 1.0/UDM                                                   KFIM
      CALL  VADD1(BBDM,BDRM,VEC12,12,1)                              KFIM
      CALL  VSCL1(QB1,VEC12,BKDM,12,1)                               KFIM
      CALL  VSCL1(UD1,HM,VEC12,12,1)                                 KFIM
      CALL  VSUB1(BKDM,VEC12,DBDM,12,1)                              KFIM
      CALL  VSCL1(ETA,DBDM,VEC12,12,1)                               KFIM
```

```
      CALL VSUB1(BKDM,VEC12,VEC13,12,1)                              KFIM 118
      CALL VSCL1(DYMP,VEC13,VEC12,12,1)                              KFIM 119
      CALL VADD1(XDS,VEC12,XDS,12,1)                                 KFIM 120
      CALL VSCL1(0.,VEC13,VEC13,36)                                  KFIM 121
      CALL VSCL1(0.0,BRDM,BRDM,37)                                   KFIM 122
      CALL MTRA(P15,VEC12,6,1)                                       KFIM 123
      CALL GET(HM,VEC6,VEC1,VEC,VEC3)                                KFIM 124
      CALL MTRA(P15,VEC12,6,1)                                       KFIM 125
      CALL MPYM(VEC12,VEC6,VEC13,1,6,1)                              KFIM 126
      CALL VADD1(VEC13,BRDM,BRDM,1,1)                                KFIM 127
      CALL MTRA(P16,VEC12,6,2)                                       KFIM 128
      CALL MPYM(VEC12,VEC6,VEC13,1,2,1)                              KFIM 129
      CALL VADD1(BRDM,VEC13,BRDM,2,2)                                KFIM 130
      CALL MTRA(P17,VEC12,6,4)                                       KFIM 131
      CALL MPYM(VEC12,VEC6,VEC13,4,6,1)                              KFIM 132
      CALL VADD1(BRDM,VEC13,BRDM,4,4)                                KFIM 133
      CALL VADD1(BDRM,BRDM,VEC13,7,1)                                KFIM 134
      CALL VSCL1(QB1,VEC13,BKRM,7,1)                                 KFIM 135
      CALL VSCL1(UD1,RM,VEC12,7,1)                                   KFIM 136
      CALL VSUB1(BKRM,VEC12,BDRM,7,1)                                KFIM 137
      CALL VSCL1(ETA,BDRM,VEC12,7,1)                                 KFIM 138
      CALL VSUB1(BKRM,VEC12,VEC13,7,1)                               KFIM 139
      CALL VSCL1(DYMP,VEC13,VEC12,7,1)                               KFIM 140
      CALL VADD1(RM,VEC12,XRS,7,1)                                   KFIM 141
      CALL VSCL1(QB,BKRM,BKRM,7,1)                                   KFIM 142
      CALL VSCL1(ETB,BRDM,BRDM,7,1)                                  KFIM 143
      CALL GET(BKRM,VEC,VEC1,VEC2,VEC3)                              KFIM 144
      CALL GET(BRDM,VEC4,VEC5,VEC6,VEC7)                             KFIM 145
      QALT=VEC(1)-VEC4(4)*QB                                         KFIM 146
      CALL VSUB1(P55,QALT,P55,1,1)                                   KFIM 147
      CALL VCXR(VEC1,VEC1,VEC12,2,2)                                 KFIM 148
      CALL VCXR(VEC5,VEC5,VEC13,2,2)                                 KFIM 149
      CALL SUBM(VEC12,VEC13,VEC14,2,2)                               KFIM 150
      CALL SUBM(P66,VEC14,P66,2,2)                                   KFIM 151
      CALL VCXR(VEC2,VEC2,VEC12,4,4)                                 KFIM 152
      CALL VCXR(VECL,VEC6,VEC13,4,4)                                 KFIM 153
      CALL SUBM(VEC12,VEC13,VEC14,4,4)                               KFIM 154
      CALL SUBM(P77,VEC14,P77,4,4)                                   KFIM 155
      CALL GET(BKDM,VEC4,VEC5,VEC6,VEC7)                             KFIM 156
```

```
QALT=VEC(1)                                               KFIM 157
CALL VSCL1(QALT,VEC4,VEC12,6,1)                           KFIM 158
CALL VSUB1(P15,VEC12,P15,6,1)                             KFIM 159
CALL MTRA(P15,P51,6,1)                                    KFIM 160
CALL VCXR(VEC4,VEC1,VEC12,6,2)                            KFIM 161
CALL SUBM(P16,VEC12,P16,6,2)                              KFIM 162
CALL MTRA(P16,P61,6,2)                                    KFIM 163
CALL VCXR(VEC4,VEC2,VEC12,6,4)                            KFIM 164
CALL SUBM(P17,VEC12,P17,6,4)                              KFIM 165
CALL MTRA(P17,P71,6,4)                                    KFIM 166
CALL VSCL1(QALT,VEC5,VEC12,3,1)                           KFIM 167
CALL VSUB1(P35,VEC12,P35,3,1)                             KFIM 168
CALL MTRA(P35,P53,3,1)                                    KFIM 169
CALL VCXR(VEC5,VEC1,VEC12,3,2)                            KFIM 170
CALL SUBM(P36,VEC12,P36,3,2)                              KFIM 171
CALL MTRA(P36,P63,3,2)                                    KFIM 172
CALL VCXR(VEC5,VEC2,VEC12,3,4)                            KFIM 173
CALL SUBM(P37,VEC12,P37,3,4)                              KFIM 174
CALL MTRA(P37,P73,3,4)                                    KFIM 175
CALL VSCL1(QALT,VEC6,VEC12,3,1)                           KFIM 176
CALL VSUB1(P45,VEC12,P45,3,1)                             KFIM 177
CALL MTRA(P45,P54,3,1)                                    KFIM 178
CALL VCXR(VEC6,VEC1,VEC12,3,2)                            KFIM 179
CALL SUBM(P46,VEC12,P46,3,2)                              KFIM 180
CALL MTRA(P46,P64,3,2)                                    KFIM 181
CALL VCXR(VEC6,VEC2,VEC12,3,4)                            KFIM 182
CALL SUBM(P47,VEC12,P47,3,4)                              KFIM 183
CALL MTRA(P47,P74,3,4)                                    KFIM 184
CALL VSCL1(QALT,P56,VEC12,2,1)                            KFIM 185
CALL VSUB1(P56,VEC12,P56,2,1)                             KFIM 186
CALL MTRA(P56,P65,2,1)                                    KFIM 187
CALL VSCL1(QALT,P57,VEC12,4,1)                            KFIM 188
CALL VSUB1(P57,VEC12,P57,4,1)                             KFIM 189
CALL MTRA(P57,P75,4,1)                                    KFIM 190
CALL VCXR(VE?2,VEC2,VEC12,2,4)                            KFIM 191
CALL VSUB1(P67,VEC12,P67,2,4)                             KFIM 192
CALL MTRA(P67,P76,2,4)                                    KFIM 193
CALL VSCL1(QB,BKDM,BKDM,12,1)                             KFIM 194
ETB = ETA*ETA                                             KFIM 195
```

```
      CALL VSCL1(ETB,DBDM,DBDM,12,1)
      CALL GET(BKDM,VEC6,VEC1,VEC,VEC5)
      CALL VCXR(VEC6,VEC6,VEC12,6,6)
      CALL GET(DBDM,VEC7,VEC8,VEC9,VEC10)
      CALL VCXR(VEC7,VEC7,VEC13,6,6)
      CALL SUBM(VEC12,VEC13,VEC14,6,6)
      CALL SUBM(P11 ,VEC14,P11 ,6,6)
      CALL VCXR(VEC6,VEC1,VEC12,6,3)
      CALL VCXR(VEC7,VEC8,VEC13,6,3)
      CALL SUBM(VEC12,VEC13,VEC14,6,3)
      CALL SUBM(P13 ,VEC14,P13 ,6,3)
      CALL MTRA(P13,P31,6,3)
      CALL VCXR(VEC6,VEC,VEC12,6,3)
      CALL VCXR(VEC7,VEC9,VEC13,6,3)
      CALL SUBM(VEC12,VEC13,VEC14,6,3)
      CALL SUBM(P14,VEC14,P14,6,3)
      CALL MTRA(P14,P41,6,3)
      CALL VCXR(VEC1,VEC1,VEC12,3,3)
      CALL VCXR(VEC8,VEC8,VEC13,3,3)
      CALL SUBM(VEC12,VEC13,VEC14,3,3)
      CALL SUBM(P33,VEC14,P33,3,3)
      CALL VCXR(VEC1,VEC,VEC12,3,3)
      CALL VCXR(VEC8,VEC9,VEC13,3,3)
      CALL SUBM(VEC12,VEC13,VEC14,3,3)
      CALL SUBM(P34,VEC14,P34,3,3)
      CALL MTRA(P34,P43,3,3)
      CALL VCXR(VEC,VEC,VEC12,3,3)
      CALL VCXR(VEC9,VEC9,VEC13,3,3)
      CALL SUBM(VEC12,VEC13,VEC14,3,3)
      CALL SUBM(P44,VEC14,P44,3,3)
  300 CONTINUF
      RETURN
C     FND OF KFIM (SUBROUTINE)
C
C     START OF KMCM (SUBROUTINE)
      FNTRY KMCM
      RFTURN
C     END OF KMCM (SUBROUTINF)
```

```
C                                                                      KMOM   1
C      START OF KMOM (SUBROUTINE)                                      KMOM   2
       ENTRY KMOM                                                      KMOM   3
       RETURN                                                          KMOM   4
C      END OF KMOM (SUBROUTINE)                                        KMOM   5
C      START OF KFCW (SUBROUTINE)                                      KFCW   1
C                                                                      KFCW   2
       ENTRY KFCW                                                      KFCW   3
       CALL VSCL1(0.0,VEC12,VEC12,36,1)                                KFCW   4
       CALL MXMV (TSW,VEC12,9)                                         KFCW   5
       CALL VSCL1(1.0,TSW,VEC12,9,28)                                  KFCW   6
       CALL ARRY (VEC12,VEC13,1)                                       KFCW   7
C      VEC13 =(TSW P) 6X6                                              KFCW   8
       CALL MPYM (VEC13,XDS,VEC12,6,6,1)                               KFCW   9
       CALL MXMV (VEC12,XDS,6)                                         KFCW  10
       CALL MPYM (VEC13,UDS,VEC12,6,6,1)                               KFCW  11
       CALL MXMV (VEC12,UDS,6)                                         KFCW  12
       CALL MPYM (VEC13,P13,VEC12,6,6,3)                               KFCW  13
       CALL MXMV (VEC12,P13,18)                                        KFCW  14
       CALL MPYM (VEC13,P14,VEC12,6,6,3)                               KFCW  15
       CALL MXMV (VEC12,P14,18)                                        KFCW  16
       CALL MXMV (P14,P41,18)                                          KFCW  17
       CALL MXMV (P13,P31,18)                                          KFCW  18
       CALL MPYM (VEC13,P15,VEC12,6,6,1)                               KFCW  19
       CALL MXMV (VEC12,P15,6)                                         KFCW  20
       CALL MXMV (VEC12,P51,6)                                         KFCW  21
       CALL MPYM (VEC13,P16,VEC12,6,6,2)                               KFCW  22
       CALL MXMV (VEC12,P16,12)                                        KFCW  23
       CALL MXMV (VEC12,P61,12)                                        KFCW  24
       CALL MTRA (VEC13,VEC12,6,6)                                     KFCW  25
C      VEC12 =(TSW P T)  6X6                                           KFCW  26
       CALL MPYM (P35,VEC12,VEC14,3,3,1)                               KFCW  27
       CALL MXMV (VEC14,P35,3)                                         KFCW  28
       CALL MXMV (VEC14,P53,3)                                         KFCW  29
       CALL MPYM (P36,VEC12,VEC14,3,3,2)                               KFCW  30
       CALL MXMV (VEC14,P36,6)                                         KFCW  31
       CALL MXMV (VEC14,P63,6)                                         KFCW  32
       CALL MPYM (P37,VEC12,VEC14,3,3,4)                               KFCW  33
```

```
      CALL MXMV (VEC14,P37,12)                                          KFCW  34
      CALL MXMV (VEC14,P73,12)                                          KFCW  35
      CALL MPYM (P45,VEC12,VEC14,3,3,1)                                 KFCW  36
      CALL MXMV (VFC14,P45,3)                                           KFCW  37
      CALL MXMV (VEC14,P54,3)                                           KFCW  38
      CALL MPYM (P46,VEC12,VEC14,3,3,2)                                 KFCW  39
      CALL MXMV (VEC14,P46,6)                                           KFCW  40
      CALL MXMV (VEC14,P64,6)                                           KFCW  41
      CALL MPYM (P47,VEC12,VEC14,3,3,4)                                 KFCW  42
      CALL MXMV (VEC14,P47,12)                                          KFCW  43
      CALL MXMV (VEC14,P74,12)                                          KFCW  44
      CALL MPYM (P11,VEC12,VEC14,6,6,6)                                 KFCW  45
      CALL MPYM (VEC13,VEC14,P11,6,6,6)                                 KFCW  46
      CALL MPYM (P17,VEC12,VEC14,6,6,4)                                 KFCW  47
      CALL MPYM (VEC13,VEC14,P17,6,6,4)                                 KFCW  48
      CALL MXMV (P17,P71,24)                                            KFCW  49
      KFFN = 2                                                          KFCW  50
      RETURN                                                            KFCW  51
C     END OF KFCW (SUBROUTINE)                                          KFCW  52
C                                                                       KMRM   1
C     START OF KMRM (SUBROUTINE)                                        KMRM   2
      ENTRY KMRM                                                        KMRM   3
      RETURN                                                            KMRM   4
      END                                                               KMRM   5
C     END OF KMRM (SUBROUTINE)                                          KMRM   6
      SUBROUTINE PO1(A,D,B,C,F)                                         KMRM   7
      DIMENSION A(1),D(1),B(1),A1(4),A2(4),A3(4)                        KMRM   8
C           ENTRY                             RETURN                    KMRM   9
C     VECTOR  A CONTAIN  G               (-G*DT + DV)                   KMRM  10
C     VECTOR  D CONTAIN  DELTA V         (-.5*G*DT*2+DP-V*DT)           KMRM  11
C     VECTOR  B CONTAIN  DELTA P         (-.1666*G*DT*2)                KMRM  12
C     VECTOR  C CONTAIN  TIME                                           KMRM  13
C     VECTOR  F CONTAIN  TOTAL V                                        KMRM  14
      DO 11 I =1,3                                                      KMRM  15
      A1(I) = A(I)                                                      KMRM  16
      A2(I) = D(I)                                                      KMRM  17
      A3(I) = B(I)                                                      KMRM  18
   11 CONTINUE                                                          KMRM  19
      DO 12 I= 1,3                                                      KMRM  20
```

```
      A(I) = A3(I) - A1(I)*C                                            KMRM  21
      D(I) = A2(I) -(.5*A1(I)*C*C)-E*C                                  KMRM  22
      B(I) = (-0.16666 *A1(I)*C*C)                                      KMRM  23
   12 CONTINUE                                                          KMRM  24
      RETURN                                                            KMRM  25
      END                                                               KMRM  26
      SUBROUTINE GET(E,A,B,C,D)                                         KMRM  27
      DIMENSION A(1),B(1),C(1),D(1)                                     KMRM  28
      K=1                                                               KMRM  29
      DO 102 I=1,6                                                      KMRM  30
      A(K)=E(I)                                                         KMRM  31
      IF (I.GT.3) GO TO 102                                             KMRM  32
      B(K)=E(I+6)                                                       KMRM  33
      C(K)=E(I+9)                                                       KMRM  34
      D(K)=E(I+6)                                                       KMRM  35
      K=K+1                                                             KMRM  36
  102 CONTINUE                                                          KMRM  37
      RETURN                                                            KMRM  38
      END                                                               KMRM  39
      SUBROUTINE ARRY(A,B,N)                                            KMRM  40
      DIMENSION A(1,1),B(1)                                             KMRM  41
      K=1                                                               KMRM  42
      DO 10 J=1,3                                                       KMRM  43
      DO 11 I=1,3                                                       KMRM  44
      A(I,J)=B(K)                                                       KMRM  45
      A(I+3,J)=B(K+9)                                                   KMRM  46
      K=K+1                                                             KMRM  47
   11 CONTINUE                                                          KMRM  48
      K=K+3                                                             KMRM  49
   10 CONTINUE                                                          KMRM  50
      GO TO (12,13),N                                                   KMRM  51
C     N=1 IS 6*6,                                                       KMRM  52
   13 CONTINUE                                                          KMRM  53
      RETURN                                                            KMRM  54
   12 CONTINUE                                                          KMRM  55
      K=19                                                              KMRM  56
      DO 14 J=4,6                                                       KMRM  57
      DO 15 I=1,3                                                       KMRM  58
      A(I,J)=B(K)                                                       KMRM  59
```

```
      A(I+3,J)=B(K+9)                                                   KMRM  60
      K=K+1                                                             KMRM  61
   15 CONTINUE                                                          KMRM  62
      K=K+3                                                             KMRM  63
   14 CONTINUE                                                          KMRM  64
      RETURN                                                            KMRM  65
      END                                                               KMRM  66
C     END OF KALMAN SUBROUTINE GROUP                                    KMRM  67
C                                                                       SPCL   1
C     SPECIAL MODULE GROUP                                              SPCL   2
C     START OF SUBROUTINE SPECIAL                                       SPCL   3
      SUBROUTINE SPECL                                                  SPCL   4
C     (TO RUN PROGRAM,COMMON BLOCK DATA AT END OF THIS LISTING          SPCL   5
C     MUST BE INSERTED)                                                 SPCL   6
      DIMENSION PRNT(10)                                                SPCL   7
      LINE=50                                                           SPCL   8
      CALL PAGE (LINE,DRMM,1,2)                                         SPCL   9
      DO 20 I=1,10                                                      SPCL  10
      PRNT(I)=0.                                                        SPCL  11
   20 CONTINUE                                                          SPCL  12
      RETURN                                                            SPCL  13
C                                                                       OUTM   1
C     START OF OUTM (SUBROUTINE)                                        OUTM   2
      ENTRY OUTM                                                        OUTM   3
      RETURN                                                            OUTM 141
      END                                                               OUTM 142
C     END OF OUTM (SUBROUTINE)                                          OUTM 143
C     FND OF SUBROUTINE SPECIAL                                         OUTM 144
C                                                                       SUBS   1
C     START OF COMMON SUBROUTINE GROUP                                  SUBS   2
C     MATRIX OPERATIONS  1                                              SUBS   3
C     SUBROUTINE ADDM,SUBM,MPYM AND TRANSPOSE MATRIX (GENERAL)          SUBS   4
                                                                        SUBS   5
C     CALLING SEQUENCE            WHERE                                 SUBS   6
C      CALL ADDM (A,B,R,N,M)       A = 1-ST.MATRIX                      SUBS   7
C      CALL SUBM (A,B,R,N,M)       B = 2-ND.MATRIX                      SUBS   8
C      CALL MPYM (A,B,R,N,M,L)     R = OUTPUT MATRIX                    SUBS   9
C      CALL MTRA (A,R,N,M)         N = NO.ROWS IN A                     SUBS  10
```

```
C                                 M = NO.CCLS IN A AND ROWS IN B        SUBS  11
C                                 L = NO.COLS IN B                      SUBS  12
      SUBROUTINE MPYM (A,B,R,N,M,L)                                     SUBS  13
      DIMENSION  A(1),B(1),R(1)                                         SUBS  14
      IR = 0                                                            SUBS  15
      IK = -M                                                           SUBS  16
      DO 10 K=1,L                                                       SUBS  17
      IK = IK+M                                                         SUBS  18
      DO 10 J=1,N                                                       SUBS  19
      IR = IR+1                                                         SUBS  20
      JI = J-N                                                          SUBS  21
      IB = IK                                                           SUBS  22
      R(IR) = 0.                                                        SUBS  23
      DO 10 I=1,M                                                       SUBS  24
      JI = JI+N                                                         SUBS  25
      IB = IB+1                                                         SUBS  26
      IF (A(JI).EQ.0.0.OR.B(IB).EQ.0.0) GO TO 10                        SUBS  27
      R(IR) = R(IR) + A(JI)* B(IB)                                      SUBS  28
   10 CONTINUE                                                          SUBS  29
      RETURN                                                            SUBS  30
      ENTRY ADDM (A,B,R,N,M)                                            SUBS  31
      K = 0                                                             SUBS  32
      GO TO 11                                                          SUBS  33
      FNTRY SUBM (A,B,R,N,M)                                            SUBS  34
      K = 1                                                             SUBS  35
   11 NM = N*M                                                          SUBS  36
      DO 12 I=1,NM                                                      SUBS  37
      IF (K) 13,13,14                                                   SUBS  38
   13 R(I) = A(I)+B(I)                                                  SUBS  39
      GO TO 12                                                          SUBS  40
   14 R(I) = A(I)-B(I)                                                  SUBS  41
   12 CONTINUE                                                          SUBS  42
      RETURN                                                            SUBS  43
      ENTRY MTRA (A,R,N,M)                                              SUBS  44
      IR = 0                                                            SUBS  45
      DO 15 I=1,N                                                       SUBS  46
      IJ = I-N                                                          SUBS  47
      DO 15 J=1,M                                                       SUBS  48
```

```
      IJ = IJ+N                                                       SUBS
      IR=IR+1                                                         SUBS
 15   P(IR)=A(IJ)                                                     SUBS
      RETURN                                                          SUBS
      END                                                             SUBS
C      VECTOR OPERATIONS 1                                            SUBS
C     SUBROUTINE DOT PRODUCT OF VECTORS (1X3 OR 3X1 VECTORS)          SUBS
C     CALLING SEQUENCE        OUTPUT                                  SUBS
C      Y = DOT (A,B)           ANGLE  (Y)                             SUBS
C      Y = ADOT  (A,B)         ANGLE  (Y)                             SUBS
C      Y = ADOTR (A,B)         ANGLE  (Y)                             SUBS
C      Y = FNORM (Z)           ANGLE (/Y/)                            SUBS
C      Y = VNORM (A,B)         VECTOR (B(I)                           SUBS
      FUNCTION DOT (X,Y)                                              SUBS
      DIMENSION X(3),Y(3),A(6),SUM(3),Z(3),W(3)                       SUBS
      DATA OVT/25755124.E3/                                           SUBS
 1    II = 1                                                          SUBS
      J2 = 4                                                          SUBS
      GO TO 3                                                         SUBS
      ENTRY ADOT (X,Y)                                                SUBS
      C = 57.2957796                                                  SUBS
      GO TO 2                                                         SUBS
      ENTRY ADOTR (X,Y)                                               SUBS
      C = 1.                                                          SUBS
 2    II = 2                                                          SUBS
      J2 = 1                                                          SUBS
 3    DO 4 I=1,3                                                      SUBS
      A(I) = X(I)                                                     SUBS
 4    A(I+3) = Y(I)                                                   SUBS
      GO TO 15                                                        SUBS
 5    I = I + 1                                                       SUBS
      GO TO (6,6,7,8),I                                               SUBS
 6    J1 = 1                                                          SUBS
      GO TO 16                                                        SUBS
 7    J2 = 4                                                          SUBS
      GO TO 16                                                        SUBS
 8    SCL = AMAX1(SUM(1),SUM(3))                                      SUBS
      IF (SCL.LF.OVT) GO TO 10                                        SUBS
      DO 9 I = 1,3                                                    SUBS
```

```
 9    SUM(I) = SUM(I)/SCL                                            SUBS  88
 10   SUM(1) = QATAND(SQRT(SUM(1)*SUM(3)-SUM(2)*SUM(2)),SUM(2))      SUBS  89
      GO TO 18                                                       SUBS  90
      ENTRY FNORM(Z)                                                 SUBS  91
      II = 3                                                         SUBS  92
      GO TO 11                                                       SUBS  93
      ENTRY VNORM (Z,W)                                              SUBS  94
      II = 4                                                         SUBS  95
 11   J2 = 1                                                         SUBS  96
      DO 12 I=1,3                                                    SUBS  97
 12   A(I) = Z(I)                                                    SUBS  98
      GO TO 15                                                       SUBS  99
 13   SUM (1) = SQRT(SUM(1))                                         SUBS 100
      IF (II.LE.3) GO TO 18                                          SUBS 101
      DO 14 I=1,3                                                    SUBS 102
 14   W(I) = A(I)/SUM(1)                                             SUBS 103
      GO TO 18                                                       SUBS 104
 15   I = 1                                                          SUBS 105
      J1 = 1                                                         SUBS 106
 16   SUM(I) = 0.                                                    SUBS 107
      DO 17 J = 1,3                                                  SUBS 108
      SUM(I) = A(J1)*A(J2)+SUM(I)                                    SUBS 109
      J1 = J1+1                                                      SUBS 110
 17   J2 = J2+1                                                      SUBS 111
      GO TO (18,5,13,13),II                                          SUBS 112
 18   DOT = SUM(1)                                                   SUBS 113
      RETURN                                                         SUBS 114
      END                                                            SUBS 115
C      VECTOR OPERATION 2                                            SUBS 116
C      SUBROUTINE CROSS PRODUCT OF VECTORS; SUM,DIFF,ETC.            SUBS 117
C                                      (3X1 OR 1X3 VECTORS)          SUBS 118
C      CALLING SEQUENCE                                              SUBS 119
C      CALL VCROS (A,B,C)     WHERE A = 1-ST VECTOR                  SUBS 120
C      CALL VADD  (A,B,C)           B = 2-ND VECTOR                  SUBS 121
C      CALL VSUB  (A,B,C)           C = RESULTANT VECTOR             SUBS 122
C      CALL VSCL  (N,A,C)           N = SCALAR OR NO.ELEMENTS IN VECTOR SUBS 123
C      CALL VNCR  (A,B,C)                                            SUBS 124
C      CALL VCXR  (A,B,C)                                            SUBS 125
C      CALL VRXC  (A,B,C)                                            SUBS 126
```

```
C      CALL VMOV  (A,C,N)
C
C
       SUBROUTINE VCROS(X,Y,Z,N,M)
       DIMENSION X(1),Y(1),Z(1)

       NP = 1
       GO TO 10
       ENTRY VADD (X,Y,Z)
       DO 20 I = 1,3
 20    Z(I) = X(I) + Y(I)
       GO TO 13
       ENTRY VSUB (X,Y,Z)
       DO 21 I = 1,3
 21    Z(I) = X(I) - Y(I)
       GO TO 13
       ENTRY VNCR (X,Y,Z)
       NP = 2
 10    Z(1) = (X(2)*Y(3))-(X(3)*Y(2))
       Z(2) = (X(3)*Y(1))-(X(1)*Y(3))
       Z(3) = (X(1)*Y(2))-(X(2)*Y(1))
       AB = SQRT (Z(1)**2+Z(2)**2+Z(3)**2)
       GO TO (13,14,15),NP
 14    DO 23 I = 1,3
 23    Z(I) = Z(I)/AB
       GO TO 13
 15    CONTINUE
       ENTRY VCXP (X,Y,Z,N,M)
       IN = 1
       DO 25 J=1,N
       DO 26 I=1,M
       Z(IN) = X(J)*Y(I)
 26    IN = IN+1
 25    CONTINUE
       GO TO 13
       ENTRY VRXC (X,Y,Z)
       Z(1) = 0
       DO 27 I = 1,3
 27    Z(1) = X(I) * Y(I) + Z(1)
```

```
      GO TO 13
   13 CONTINUE
      RETURN
      FNTRY VADD1(X,Y,Z,N,M)
      IF (M.EQ.0) M=1
      M1=M
      DO 16 I=1,N
      Z(M1)=X(I)+Y(I)
      M1=M1+1
   16 CONTINUE
      RETURN
      ENTRY VSUB1(X,Y,Z,N,M)
      IF (M.EQ.0) M=1
      M1=M
      DO 17 I=1,N
      Z(M1)=X(I)-Y(I)
      M1=M1+1
   17 CONTINUE
      RETURN
      END
C     MATRIX OPERATIONS 2

C     SUBROUTINE STRN,GTSN,GTRN
C      GENERATES THE 3X3 TRANSFORMATION MATRIX CORRESPONDING TO A
C     ROTATION ABOUT A LINE OR A SEQUENCE OF ROT. ABOUT COOR. AXES
C     CALLING SEQUENCE          WHERE ALL OPERATION ARE 3X3,3X1 OR(1X3)
C                               T = MATRIX RESULT.     (3X3)
C      CALL GTRN (T,N,A,M)      N = ROTATION AXIS
C      CALL GTSN (T,N,SA,CA,M)  M = NUMBER OF ROTATION
C      CALL STRN (T,IS)         SA,CA = SINE,COSINE
C                               A = ANGLE OF ROTATION (RAD.)  (3)
      SUBROUTINE GTRN (T,N,A,M)
      COMMON ST(9),WT(9)
      DIMENSION A(10),N(10),T(9),NN(4),NS(22),CA(10),SA(10)
      DATA NN/6,7,2,3/,NS/5,6,4,8,9,7,2,3,1,5,6,4,5,9,7,8,3,1,2,6,4,5/,
     1 MTEST,TTEST/10,5.E-8/
    1 ASSIGN 6 TO N1
      GO TO 2
      FNTRY GTSN(T,N,SA,CA,M)
```

```
      ASSIGN 7 TO N1                                                    SUBS 205
    2 MG = 1                                                            SUBS 206
      MS = 1                                                            SUBS 207
      ASSIGN 9 TO N2                                                    SUBS 208
      IF (N(1).EQ.0) GO TO 5                                            SUBS 209
      DO 3 I=1,9                                                        SUBS 210
    3 ST(I) = 0.                                                        SUBS 211
      DO 4 I = 1,9,4                                                    SUBS 212
    4 ST(I) = 1.                                                        SUBS 213
      MG = M                                                            SUBS 214
      ASSIGN 13 TO N2                                                   SUBS 215
    5 DO 15 I=1,MG                                                      SUBS 216
      GO TO N1,(6,7)                                                    SUBS 217
    6 WT(4) = A(I)                                                      SUBS 218
      C = COS(WT(4))                                                    SUBS 219
      S = SIN(WT(4))                                                    SUBS 220
      GO TO 8                                                           SUBS 221
    7 C = CA(I)                                                         SUBS 222
      S = SA(I)                                                         SUBS 223
    8 GO TO N2,(9,13)                                                   SUBS 224
    9 CALL VNORM(M,WT)                                                  SUBS 225
      L = 1                                                             SUBS 226
      WT(4)=1.0-C                                                       SUBS 227
      DO 10 J=1,3                                                       SUBS 228
      WT(5) = WT(4)*WT(J)                                               SUBS 229
      DO 10 K=1,3                                                       SUBS 230
      ST(L) = WT(5)*WT(K)                                               SUBS 231
   10 L = L+1                                                           SUBS 232
      DO 11 J=1,9,4                                                     SUBS 233
   11 ST(J)=ST(J)+C                                                     SUBS 234
      DO 12 J=1,3                                                       SUBS 235
      K = NN(J)                                                         SUBS 236
      ST(K)= -WT(J)*S + ST(K)                                           SUBS 237
   12 ST(L)=  WT(J)*S + ST(L)                                           SUBS 238
      GO TO 15                                                          SUBS 239
   13 J = IABS(N(I))+1                                                  SUBS 240
      IF (N(I).LT.0) S=-S                                               SUBS 241
      IF (J.GT.3) J=1                                                   SUBS 242
      K = J+1                                                           SUBS 243
```

```
      IF (K.GT.3) K=1                                                   SUBS 244
      DO 14 L=1,3                                                       SUBS 245
      WT(4) = ST(J)                                                     SUBS 246
      ST(J) = C*WT(4)+S*ST(K)                                           SUBS 247
      ST(K) =-S*WT(4)+C*ST(K)                                           SUBS 248
      J = J+3                                                           SUBS 249
   14 K = K+3                                                           SUBS 250
   15 CONTINUE                                                          SUBS 251
      IF (MG.GT.MTEST) GO TO 18                                         SUBS 252
      DO 16 I=1,9                                                       SUBS 253
   16 T(I) = ST(I)                                                      SUBS 254
      RETURN                                                            SUBS 255
      ENTRY STRN (T,IS)                                                 SUBS 256
      MS = IS                                                           SUBS 257
      DO 17 I=1,9                                                       SUBS 258
   17 ST(I)= T(I)                                                       SUBS 259
   18 IF (MS.EQ.0) GO TO 21                                             SUBS 260
      DO 20 I=1,MS                                                      SUBS 261
      DO 19 J=1,9                                                       SUBS 262
      K = NS(J+0)                                                       SUBS 263
      L = NS(J+10)                                                      SUBS 264
      KK = NS(J+13)                                                     SUBS 265
      LL = NS(J+3)                                                      SUBS 266
   19 WT(J) = ST(K)*ST(KK)-ST(L)*ST(LL)                                 SUBS 267
      WM = 1.0/DOT(ST,WT)                                               SUBS 268
      DO 20 J=1,9                                                       SUBS 269
   20 ST (J) = (WT(J)*WM+ST(J))*0.5                                     SUBS 270
   21 CONTINUE                                                          SUBS 271
      DO 22 J = 1,9                                                     SUBS 272
      IF (ABS(ST(J)).LE.TTEST) ST(J)=0.0                                SUBS 273
   22 T(J) = ST(J)                                                      SUBS 274
      RETURN                                                            SUBS 275
      END                                                               SUBS 276
C                                                                       SUBT   1
C     MATRIX OPERATION 3                                                SUBT   2
C     SUBROUTINE MTRN,MTRT,VTRN,VTRT,WCROS,MXMV,MGID,MSCL               SUBT   3
C     CALLING SEQUENCE        WHERE ALL OPERATIONS ARE 3X3,3X1 OR 1X3   SUBT   4
C      CALL MTRN (A,B,C)      A = 1-ST MATRIX        NOTE:(3X3)*(3X3)   SUBT   5
```

```
C       CALL MTRT (A,B,C)       B = 2-ND MATRIX OR VECTOR    (3X3)*(3X3)T SUBT   6
C       CALL VTRN (A,B,C)       C = RESULT                   (3X3)*(3X1)  SUBT   7
C       CALL VTRT (A,B,C)                                    (3X3)T*(3X1) SUBT   8
C       CALL WCROS (A,B)                                                  SUBT   9
      SUBROUTINE MTRN (A,B,C)                                             SUBT  10
      DIMENSION A(1),B(1),C(1),A1(1,1,1)                                  SUBT  11
      M1=3                                                                SUBT  12
      M2=8                                                                SUBT  13
      N=3                                                                 SUBT  14
      GO TO 1                                                             SUBT  15
      ENTRY MTRT(A,B,C)                                                   SUBT  16
      M1=1                                                                SUBT  17
      M2=0                                                                SUBT  18
      N=3                                                                 SUBT  19
      GO TO 1                                                             SUBT  20
      ENTRY VTRN(A,B,C)                                                   SUBT  21
      M1=3                                                                SUBT  22
      M2=8                                                                SUBT  23
      N=1                                                                 SUBT  24
      GO TO 1                                                             SUBT  25
      ENTRY VTRT(A,B,C)                                                   SUBT  26
      M1=1                                                                SUBT  27
      M2=0                                                                SUBT  28
      N=1                                                                 SUBT  29
    1 I1=1                                                                SUBT  30
      J1=1                                                                SUBT  31
      K1=1                                                                SUBT  32
      DO 4 I=1,N                                                          SUBT  33
      DO 3 J=1,3                                                          SUBT  34
      CI1 = 0.0                                                           SUBT  35
      DO 2 K=1,3                                                          SUBT  36
      CI1= A(J1)*B(K1)+CI1                                                SUBT  37
      J1=J1+M1                                                            SUBT  38
    2 K1=K1+1                                                             SUBT  39
      C(I1) = CI1                                                         SUBT  40
      I1=I1+1                                                             SUBT  41
      J1=J1-M2                                                            SUBT  42
```

```
3     K1=K1-3                                                    SUBT   43
      K1=K1+3                                                    SUBT   44
4     J1=1                                                       SUBT   45
      RETURN                                                     SUBT   46
      ENTRY WCROS (A,B)                                          SUBT   47
      B(1) = 0.                                                  SUBT   48
      B(4)=-A(3)                                                 SUBT   49
      B(7)=A(2)                                                  SUBT   50
      B(2)=A(3)                                                  SUBT   51
      B(5) = 0.0                                                 SUBT   52
      B(8)=-A(1)                                                 SUBT   53
      B(3)=-A(2)                                                 SUBT   54
      B(6)=A(1)                                                  SUBT   55
      B(9) = 0.0                                                 SUBT   56
      RETURN                                                     SUBT   57
      END                                                        SUBT   58
      SUBROUTINE NORN (S,AM,IX,V)                                SUBT   59
      A=0.                                                       SUBT   60
      DO 2 I=1,12                                                SUBT   61
      IY=IX*65539                                                SUBT   62
      IF (IY.GE.0) GU TO 1                                       SUBT   63
      IY=IY+2147483647+1                                         SUBT   64
1     Y=IY                                                       SUBT   65
      Y=Y*.4656613E-9                                            SUBT   66
      IX=IY                                                      SUBT   67
2     A=A+Y                                                      SUBT   68
      V=(A-6.)*S+AM                                              SUBT   69
      RETURN                                                     SUBT   70
      END                                                        SUBT   71
      SUBROUTINE INVERT (A,N,M,D)                                SUBT   72
      DIMENSION A(1)                                             SUBT   73
      COMMON T,S,ND,DT,NM,NI,IN(20)                              SUBT   74
1     DT=1.0                                                     SUBT   75
      NI=N                                                       SUBT   76
      NM=M                                                       SUBT   77
      IF (NI) 22,22,2                                            SUBT   78
2     ND=NM-NI                                                   SUBT   79
      NN=NI*NM                                                   SUBT   80
      DO 3 J=1,NI                                                SUBT   81
```

```
3      IN(J)=0                                            SUBT  82
       DO 15 L=1,NI                                       SUBT  83
       T=0.0                                              SUBT  84
       K=1                                                SUBT  85
       DO 9 J=1,NI                                        SUBT  86
       DO 5 I=1,NI                                        SUBT  87
       IF (IN(I)-J) 5,4,5                                 SUBT  88
4      K=K+NM                                             SUBT  89
       GO TO 9                                            SUBT  90
5      CONTINUE                                           SUBT  91
       DO 8 I=1,NI                                        SUBT  92
       IF (IN(I)) 8,6,8                                   SUBT  93
6      IF (ABS(A(K))-ABS(T)) 8,8,7                        SUBT  94
7      ICOL=J                                             SUBT  95
       IROW=I                                             SUBT  96
       K1=K                                               SUBT  97
       T=A(K)                                             SUBT  98
8      K=K+1                                              SUBT  99
       K=K+ND                                             SUBT 100
9      CONTINUE                                           SUBT 101
       IN(IROW)=ICOL                                      SUBT 102
       A(K1)=1.0                                          SUBT 103
       DT=T*DT                                            SUBT 104
       IF (DT) 10,22,10                                   SUBT 105
10     K=K1-IROW                                          SUBT 106
       DO 13 I=1,NI                                       SUBT 107
       K=K+1                                              SUBT 108
       IF (IROW-I) 11,13,11                               SUBT 109
11     S=-A(K)/T                                          SUBT 110
       A(K)=0.0                                           SUBT 111
       J1=I                                               SUBT 112
       DO 12 J=IROW,NN,NM                                 SUBT 113
       A(J1)=A(J)*S+A(J1)                                 SUBT 114
12     J1=J1+NM                                           SUBT 115
13     CONTINUE                                           SUBT 116
       DO 14 J=IROW,NN,NM                                 SUBT 117
14     A(J)=A(J)/T                                        SUBT 118
15     CONTINUE                                           SUBT 119
       DO 21 I=1,NI                                       SUBT 120
```

Given a vector b,

$$\delta|b| = \frac{b^T}{|b|}\delta b, \text{ where } \frac{b}{|b|} \text{ is a unit vector pointing in the direction of b.} \qquad (140)$$

$$\frac{d}{dt}|b| = \frac{b^T}{|b|}\frac{d(b)}{dt} \qquad (141)$$

$$\delta\left\{\frac{b}{|b|}\right\} = \left\{I - \frac{b}{|b|}\frac{b^T}{|b|}\right\}\frac{\delta b}{|b|} \; ; \; I \triangleq \text{Identity Matrix} \qquad (142)$$

Proof of (140):

$$|b|\,|b| = |b|^2 = b^T b, \text{ which implies that}$$

$$\delta\{|b|\,|b|\} = \delta\{b^T b\}$$

Evaluating both sides yields:

$$2|b|\,\delta|b| = 2b^T\delta b \Rightarrow \delta|b| = \frac{b^T}{|b|}\delta b$$

Proof of (141):

$$\frac{d}{dt}|b|^2 = \frac{d}{dt}(b^T b)$$

Again evaluating both sides,

$$|b|\frac{d}{dt}|b| = b^T\frac{d(b)}{dt} \Rightarrow \frac{d|b|}{dt} = \frac{b^T}{|b|}\frac{d}{dt}(b)$$

Proof of (142):

$$\delta\left\{\frac{b}{|b|}\right\} = \frac{|b|\,\delta(b) - (b)\,\delta|b|}{|b|\,|b|}$$

Substituting in the results of (140) above yields

$$\delta\left\{\frac{b}{|b|}\right\} = \frac{|b|\,\delta(b)}{|b|\,|b|} - \left(\frac{b\,b^T}{|b|\,|b|}\right)\frac{\delta b}{|b|} = \left\{I - \frac{b\,b^T}{|b|\,|b|}\right\}\frac{\delta b}{|b|}$$

```
      GO TO 12
16    CONTINUE
      RETURN
      END
      SUBROUTINE VMOV(A,B,N,M)
      DIMENSION A(1),B(1)
      GO TO 10
      ENTRY MXMV(A,B,N)
10    I=1
      DO 11 I=1,N
      B(I)=A(I)
11    CONTINUF
      RETURN
      ENTRY MPVC(A,B,N,M)
      NS=N-1
      NZ=NS*NS
      K=M
      DO 9 I=1,NZ,N
      A(I)=A(I)+B(K)
      K=K+1
9     CONTINUE
      RETURN
      END
      SUBROUTINE MXMV1(A,B,L,M,N,DS,N1)
      DIMENSION A(1),B(1,1,1),DS(1,1,1)
      DO 8 I=1,L
8     B(I,M,N)=A(I)
      RETURN
      ENTRY MXMV2(A,B,L,M,N,DS)
      DO 7 I=1,L
7     A(I)=B(I,M,N)
      RETURN
      ENTRY MXMV3(A,B,L,M,N,DS)
      DO 10 L1=1,L
      B(L1,M,N) = B(L1,M,N)+A(L1)
10    CONTINUE
      RETURN
      ENTRY MXMV4 (A,B,L,M,N,DS)
      B(1,M,1) = 1.0 + B(1,M,1)
```

```
      B(2,M,1)=(DS(1,2,N)-DS(1,9,N))+B(2,M,1)
      B(3,M,1) = DS(1,2,N)+B(3,M,1)
      B(4,M,1) = DS(1,9,N)+B(4,M,1)
      B(5,M,1) = DS(1,8,N)+B(5,M,1)
      B(6,M,1) = DS(1,10,N)+B(6,M,1)
      RETURN
      ENTRY MXMV5(A,B,L,M,N,DS)
      DO 11 J= 1,6
      IF (B(1,J,1).EQ.0.0) GO TO 11
      J1 = J
      DO 12 K=1,4
      DO 12 I = 3,50
      K1 = K
      IF (B (I,J1,K).EQ.0.0) GO TO 11
      B(I,J1,K) = B(I,J1,K)/B(1,J,1)
   12 CONTINUE
      B(2,J,K1) = B(3,J,K1) - B(4,J,K1)
   11 CONTINUE
      RETURN
      ENTRY MXMV6(A,B,L,M,N,DS,N1)
      N2=N1
      DO 13 L1=1,L
      B(N2,M,N)=B(N2,M,N)+A(L1)
      N2=N2+1
   13 CONTINUE
      RETURN
      END
      SUBROUTINE MGIO(A,N)
      DIMENSION A(1)
      NZ=N-1
      NS=NZ*NZ
      DO 7 I=1,NS
    7 A(I)=0.0
      DO 8 I=1,NS,N
      A(I)=1.0
    8 CONTINUE
      RETURN
C     END OF COMMON SUBROUTINE GROUP
      END
```

APPENDIX I

SOME BASIC VECTOR/MATRIX RELATIONSHIPS

This appendix derives several basic, mathematical, vector/matrix relationships which are employed in the derivations in subsequent appendices.

Consider any two orthogonal frames F1 and F2. Denoting the orthogonal transformation from F1 to F2 by $T_{F2/F1}$,

$$(V_j)_{F2} = T_{F2/F1} (V_j)_{F1} \quad (j = 1,2,3) \tag{23}$$

where V_j is a unit vector along the j^{th} axis of frame F2. Taking the time rate of change of both sides of equation (23) gives

$$\dot{T}_{F2/F1} (V_j)_{F1} + T_{F2/F1} (\dot{V}_j)_{F1} = 0 \quad (j = 1,2,3) \tag{24}$$

If $\omega_{F2/F1}$ denotes the angular rate of F2 with respect to F1, it is true that:

$$(\dot{V}_j)_{F1} = (\omega_{F2/F1})_{F1} \times (V_j)_{F1} \quad (j = 1,2,3) \tag{25}$$

Substituting this result in equation (24) gives

$$[\dot{T}_{F2/F1} + T_{F2/F1} (\omega_{F2/F1})_{F1} \times] \quad V = 0 \tag{26}$$

where:

$$V = [(V_1)_{F1} \mid (V_2)_{F1} \mid (V_3)_{F1}] = T_{F1/F2} [(V_1)_{F2} \mid (V_2)_{F2} \mid (V_3)_{F2}] = T_{F1/F2}$$

and $\{(\omega_{F2/F1})_{F1}\times\}$ is the matrix equivalent of the vector cross-product operation; i.e.,

$$\{(\omega_{F2/F1})_{F1}\times\} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

where ω_1, ω_2, ω_3 are the components of $\omega_{F2/F1}$ in the F1 frame.

Since $V = T_{F1/F2}$ is invertible, it follows by postmultiplication of equation 4) by $V^{-1} = T_{F2/F1}$ that:

$$\boxed{\dot{T}_{F2/F1} = -T_{F2/F1}\,[(\omega_{F2/F1})_{F1}\times]} \tag{27}$$

Equation (27) is the first of the desired relationships. It relates the time rate of change of an orthogonal matrix to the matrix itself and to the angular rate vector as shown.

A second useful relationship is obtained from:

$$T_{F1/F2}\,T_{F2/F1} = I \tag{28}$$

Taking the time rate of change of this equation, and postmultiplying the result by $T_{F1/F2}$, leads to the desired second relationship:

$$\boxed{\dot{T}_{F1/F2} = -\,T_{F1/F2}\,\dot{T}_{F2/F1}\,T_{F1/F2}} \tag{29}$$

Equation (29) simply expresses the relationship between the time rate of change of the transformation $T_{F2/F1}$ and that of its inverse $T_{F1/F2}$.

Pre- and post multiplication of equation (27) by $T_{F1/F2}$ and use of equation (29) on the result leads to:

$$\dot{T}_{F1/F2} = [(\omega_{F2/F1})_{F1} \times] \; T_{F1/F2} \tag{30}$$

But since F1 and F2 are conceptually interchangeable, equation (30) can also be written:

$$\boxed{\dot{T}_{F2/F1} = - [(\omega_{F2/F1})_{F2} \times] \; T_{F2/F1}} \tag{31}$$

where use has been made of the fact that $\omega_{F2/F1} = - \omega_{F1/F2}$. Equation (31) is an alternate, equally useful form of the relationship inherent in equation (27).

Equating the right sides of equations (27) and (31) leads to:

$$\boxed{[(\omega_{F2/F1})_{F2} \times] = T_{F2/F1} \; [(\omega_{F2/F1})_{F1} \times] \; T_{F1/F2}} \tag{32}$$

Equation (32) is a formula for transforming the angular rate cross-product matrix from one orthogonal frame to another. Still another useful relationship, known as Coriolis' law, is obtained as follows from:

$$(a)_{F2} = T_{F2/F1} \; (a)_{F1} \tag{33}$$

where a is any vector. Taking the time rate of change of equation (33) gives:

$$(\dot{a})_{F2} = \dot{T}_{F2/F1} \; (a)_{F1} + T_{F2/F1} \; (\dot{a})_{F1} \tag{34}$$

Using equation (27) in equation (34) gives:

$$(\dot{a})_{F2} = T_{F2/F1} \left[(\dot{a}_{F1}) - \left\{ (\omega_{F2/F1})_{F1} \times \right\} (a)_{F1} \right] \tag{35}$$

from which it follows that:

$$T_{F1/F2} \; (\dot{a})_{F2} = (\dot{a})_{F1} - [(\omega_{F2/F1})_{F1} \times] \; (a)_{F1} \tag{36}$$

Applying the time derivative notation defined in the Symbol Glossary finally gives:

$$\boxed{\frac{d_{F2}a}{dt} = \frac{d_{F1}a}{dt} - [(\omega_{F2/F1})\,x\,]\,a} \tag{37}$$

Another formula, which is used for error equation linearization in Appendix III, is stated here without proof.*

$$T_{F2/F1} = I \cos|\theta| + (1-\cos|\theta|)\,uu^T - \sin|\theta|\,(ux) \tag{38}$$

where θ = vector representing the rotation of F1 into F2, and $u = \theta/|\theta|$ = unit vector about which the rotation angle $|\theta|$ is measured.

Two final formulae are also useful. If a and b are any (3x1) vectors, then:

$$(ax)\,(bx) - (bx)\,(ax) = -\left\{(axb)x\right\} \tag{39}$$

and:

$$(ax)\,(ax) = aa^T - a^TaI \tag{40}$$

These formulae can both easily be verified by carrying out the indicated operations at the scalar level.

*The proof requires use of matrix concepts more advanced than are pertinent to this development. See, for example, NASA Contractor Report CR-968, "A Study of the Critical Computational Problems Associated with Strapdown Navigation Systems", April 1968, (Appendix C).

APPENDIX II

GENERALIZED DR NAVIGATION EQUATIONS

This appendix derives a set of basic, generalized, vector/matrix, terrestrial DR navigation equations. These equations provided the partial basis for processor reference frame selection and for IDR, ADR, and PDR equation definitions.

The basic, terrestrial navigation acceleration equation is:

$$\frac{d_I^2 P}{dt^2} = f + G \tag{41}$$

Using Coriolis' law (see Appendix I) and the definition of v (see the Symbol Glossary) leads to:

$$\frac{d_I P}{dt} = v + (\omega_{E/I} X) P \tag{42}$$

and further:

$$\frac{d_I^2 P}{dt^2} = \frac{d_c v}{dt} + \left(\frac{d_c \omega_{E/I} X}{dt}\right) P + (\omega_{E/I} X) \frac{d_c P}{dt} + (\omega_{C/I} X) \left[v + (\omega_{E/I} X) P\right] \tag{43}$$

again using Coriolis' law, it follows that

$$\frac{d_c \omega_{E/I}}{dt} = (\omega_{E/C} X) \omega_{E/I} \tag{44}$$

and also that:

$$\frac{d_c P}{dt} = v + (\omega_{E/C} X) P \tag{45}$$

Substitution of these results into equation (43) and of that result into equation (41) leads to:

$$\boxed{\frac{d_c v}{dt} = f + g - \left[(2\,\omega_{E/I} + \omega_{C/E})X\right] v} \quad (46)$$

Where $g = G - (\omega_{E/I}X)(\omega_{E/I}X)P$

Equation (46) is the desired generalized acceleration equation, and equation (45) is the desired generalized relationship between position, velocity, and angular rate.

If the C, P, and E frames have relative angular rates with respect to one another, then it is necessary to initialize the transformations between these frames and subsequently to continuously update them according to the equations [see Equation (31) of Appendix I].

$$\dot{T}_{C/E} = -\left[(\omega_{C/E})_C X\right] T_{C/E} \quad (47)$$

$$\dot{T}_{C/P} = \left[(\omega_{P/C})_C X\right] T_{C/P} \quad (48)$$

The angular rate $(\omega_{C/E})_C$ required by equation 7) is obtained by solution of equation (45)*. The angular rate $(\omega_{P/C})_C$ required by equation (48), however, must be computed for inertial navigation from a formula derived as follows:

$$(\omega_{P/C})_C = (\omega_{P/I} - \omega_{C/E} - \omega_{E/I})_C \quad (49)$$

Therefore, if the inertial platform is strapdown:

$$\boxed{(\omega_{P/C})_C = T_{C/P}(\omega_{P/I})_P - (\omega_{C/E})_C - T_{C/E}(\omega_{E/I})_E} \quad (50)$$

---

*For the case of special interest C = L, equation (45) can be shown to reduce to $\dot{h} = v_1$ and $\omega_{L/E} = K_L v_L$, where the 3X3 earth radii of curvature matrix K_L is a function of h and certain elements of $T_{C/E}$.

where $(\omega_{P/I})_P$ is obtained (after suitable calibration corrections) essentially as the outputs of the strapdown gyros.

On the other hand, if the inertial platform is gimballed, $(\omega_{P/C})_C$ must be specified computationally**. Once this is done, then the platform gyro control torquing rates can be essentially specified as:

$$(\omega_{P/I})_P = T_{P/C}\left[(\omega_{P/C})_C + (\omega_{C/E})_C + T_{C/E}(\omega_{E/I})_E\right] \qquad (51)$$

**For example, for the case C = E and P = L, as $\omega_{P/C} = T_{C/P}K_L\, T_{P/C}v$

APPENDIX III

IDR NAVIGATION AND ERROR EQUATIONS

This appendix derives the processor IDR navigation and navigation error equations, predicated on the frame selection C = E (or EF) and P = L.

1. IDR NAVIGATION EQUATIONS

For the choice C = E, it follows that $\omega_{C/E} = 0$, and the processor acceleration and velocity equations can be written down direction from equations (46) and (45) of Appendix II as*.

$$\boxed{\begin{aligned} f_C &= T_{C/P} f_P \qquad (52) \\ \dot{v} &= f_C + g - 2(\omega_{E/I} X)\, v \qquad (53) \\ \dot{P} &= v \qquad (54) \end{aligned}}$$

Also in this case, the need for mechanizing equation (47) of Appendix II is obviated, since again, $\omega_{C/E} = 0$. On the other hand [see equation (31) of Appendix I and equation (48) of Appendix II]:

$$\boxed{\dot{T}_{P/C} = -\left[\left(\omega_{P/C}\right)_P X\right] T_{P/C}} \qquad (55)$$

is required for updating $T_{P/C}$. If the platform is strapdown, $\omega_{P/C}$ is computed for use in equation (55) [see equation (50) of Appendix II]:

$$\boxed{\left(\omega_{P/C}\right)_P = \left(\omega_{P/I}\right)_P - T_{P/C}\ \omega_{E/I}} \qquad (56)$$

*For compactness here, C-subscripting of vectors is omitted.

where $(\omega_{P/I})_P$ is obtained directly as the (compensated) strapdown gyro outputs. On the otherhand, for the gimballed platform case, the selection P = L leads to (see Appendix II footnotes):

$$\boxed{\left(\omega_{P/C}\right)_P = K_L T_{P/C} v} \tag{57}$$

and the required (uncompensated) platform control torquing rate from equation 11) of Appendix II is:

$$\boxed{\left(\omega_{P/I}\right)_P = \left(\omega_{P/C}\right)_P + T_{P/C}\ \omega_{E/I}} \tag{58}$$

To complete these equations, f_p for equation (52) is generated from the accelerometer outputs f_{ACC}, and from the accelerometer outputs calibration/shaping compensation Δf as:

$$\boxed{f_P = f_{ACC} + \Delta f} \tag{59}$$

Also, the angular rate calibration/shaping compensation $\Delta\omega$ is used to correct $(\omega_{P/I})_P$ in equations (56) and (58) so that:

$$\boxed{\omega_{GYR} = \left(\omega_{P/I}\right)_P + \Delta\omega} \tag{60}$$

where ω_{GYR} is the actual gyro outputs for the strapdown case, and the actual gyro control torquing rates for the gimballed platform case. Equations (52) through (58) comprise the set of basic mechanization equations for the case C=E.* For the case C=EF, it should first be noted that again, $\omega_{C/E} = 0$. In addition, since $\dot{C}$ is a constant vector in any earth-fixed frame, it follows that $\dot{p} = \dot{P}$. Finally, it is assumed that at the time of E to EF transition, p (= P-C), v, and $\omega_{E/I}$ are transformed into the EF frame, and $T_{P/C}$ is adjusted to account for the change in C frame, according to the equations:

*Equations (59) and (60) do not depend on the choice of the C frame, and thus are the same for C=E or C=EF.

$$
\boxed{\begin{aligned}
&a)\quad v_{EF} = T_{EF/E}\, v_E \\
&b)\quad p_{EF} = T_{EF/E}\, (P_E - C_E) \\
&c)\quad \left(\omega_{E/I}\right)_{EF} = T_{EF/E}\left(\omega_{E/I}\right)_E \\
&d)\quad T_{P/EF} = T_{P/E}T^T_{EF/E}
\end{aligned}}
\qquad (61)
$$

In these equations, it should be noted that $T_{EF/E}$ is a fixed transformation. With these preliminaries, it is evident that use of the generalized equations of Appendix II for the case C = EF will lead to a set of mechanization equations which is functionally identical to those above for the case C=E*, except that:

a) Equation (54) is replaced by:

$$
\boxed{\dot{p} = v} \qquad (62)
$$

b) The E frame computation of gravity, $g = g_E(P_E)$ must be replaced by the computation:

$$
\boxed{g_{EF} = T_{EF/E}\, g_E\left(T_{E/EF}\, p_{EF} + C_E\right)} \qquad (63)
$$

Finally, in EF to E transition, equations (52) through (58) will subsequently be valid if the following switching operations are carried out:

$$
\boxed{\begin{aligned}
&a)\quad v_E = T^T_{EF/E}\, v_{EF} \\
&b)\quad P_E = T^T_{EF/E}\, p_{EF} + C_E \\
&c)\quad \left(\omega_{E/I}\right)_E = T^T_{EF/E}\left(\omega_{E/I}\right)_{EF} \\
&d)\quad T_{P/E} = T_{P/EF}T_{EF/E}
\end{aligned}}
\qquad (64)
$$

---

* It should be recalled that all unsubscripted quantities in equations (52) through (60) and all time derivatives are referred to the C frame, be it E or EF.

2. IDR NAVIGATION ERROR EQUATIONS

A fundamental preliminary to the development of the inertial navigation equations (for the selections, C = E or EF, and P = L or strapdown) consists in the identification of two different sets of navigation equations, the differencing of which leads directly to the desired error equations.

The first of these two sets describes the computational process which is actually carried out, as follows:

$$\hat{f}_C = \hat{T}_{C/P}\,(f_{ACC} + \hat{\Delta f}) \tag{65}$$

$$\dot{\hat{v}} = \hat{f}_C + \hat{g}(\hat{P}) - 2(\omega_{E/I} X)\,\hat{v} \tag{66}$$

$$\dot{\hat{P}} = \dot{\hat{p}} = \hat{v} \tag{67}$$

$$\dot{\hat{T}}_{P/C} = -\left[\left(\hat{\omega}_{P/C}\right)_P X\right] \hat{T}_{P/C} \qquad \left(\begin{array}{l}\text{EITHER TYPE}\\ \text{OF PLATFORM}\end{array}\right) \tag{68}$$

$$\left(\hat{\omega}_{P/C}\right)_P = \omega_{GYR} - \hat{\Delta\omega} - \hat{T}_{P/C}\,\omega_{E/I} \qquad \left(\begin{array}{l}\text{STRAPDOWN}\\ \text{PLATFORM}\end{array}\right) \tag{69}$$

$$= \hat{K}_L \hat{T}_{P/C} \hat{v} \qquad \left(\begin{array}{l}\text{LOCAL LEVEL,}\\ \text{WANDER AZIMUTH}\\ \text{PLATFORM}\end{array}\right) \tag{70}$$

$$\omega_{GYR} = \left(\hat{\omega}_{P/C}\right)_P + \hat{T}_{P/C}\,\omega_{E/I} + \hat{\Delta\omega} \tag{71}$$

In these equations, the superior hat above a quantity indicates that it is only an onboard computational estimate of the value of the quantity, and not in general the true value. That is, these equations describe the actual onboard processing of specific force as measured by the accelerometers, into estimated velocity, position, and torquing rate signals (for the estimated platform-to-computer transformation, and, in the rotationally isolated platform case, for the platform gyros).

The second set of navigation equations is:

$$f_C = T_{C/P}\, f_P \tag{72}$$

$$\dot{v} = f_C + g(P) - 2\left(\omega_{E/I} X\right) v \tag{73}$$

$$\dot{P} = \dot{p} = V \tag{74}$$

$$\dot{T}_{P/I} = -\left[\left(\omega_{P/I}\right)_P X\right] T_{P/I} \tag{75}$$

$$\left(\omega_{P/I}\right)_P = \omega_{GYR} - \Delta\omega \tag{76}$$

(EITHER TYPE OF PLATFORM)

All quantities here are unhatted to indicate that they are true, rather than estimated values. These equations describe, a) the processing of true (as opposed to measured) specific force, resolved into the computational frame using the correct transformation between platform and computer frames, into true velocity and position, and b) the behavior of the actual, as opposed to the estimated, orientation of the platform.

The error equations can now be obtained as follows. To begin, direct differencing of equations (65) with (72), (66) with (73), and (67) with (74) gives:

$$\delta f_P = f_{ACC} + \hat{\Delta f} - f_P \tag{77}$$

$$\delta f_C = \hat{T}_{C/P}\, \delta f_P + \delta T_{C/P} f_P \tag{78}$$

$$\delta \dot{V} = \delta f_C + \delta g - 2\left(\omega_{E/I} X\right) \delta V \tag{79}$$

$$\delta \dot{P} = \delta \dot{p} = \delta V \tag{80}$$

$$\delta g = \hat{g}(\hat{P}) - g(P) \tag{81}$$

where the error quantities are defined as $\delta V = \hat{V} - V$, $\delta P = \hat{P} - P$, $\delta p = \hat{p} - p$, and $\delta T_{C/P} = \hat{T}_{C/P} - T_{C/P}$.

Noting that:

$$\hat{T}_{C/P} \equiv T_{C/\hat{P}} \tag{82}$$

it follows that:

$$\delta T_{C/P} = \hat{T}_{C/P} - T_{C/P} = T_{C/\hat{P}}\left(I - T_{\hat{P}/P}\right) \tag{83}$$

The dynamic behavior of $T_{\hat{P}/P}$ is obtained as follows. Since:

$$T_{\hat{P}/P} = T_{\hat{P}/C}\, T_{C/P} \tag{84}$$

it follows that, using equation (68), and equations (27) and (32) from Appendix I:

$$\begin{aligned}
\dot{T}_{\hat{P}/P} &= \dot{T}_{\hat{P}/C} T_{C/P} + T_{\hat{P}/C} \dot{T}_{C/P} \\
&= -\left[\left(\hat{\omega}_{P/C}\right)_P^X\right] T_{\hat{P}/C} T_{C/P} + T_{\hat{P}/P} T_{P/C} \dot{T}_{C/P} \\
&= -\left[\left(\hat{\omega}_{P/C}\right)_P^X\right] T_{\hat{P}/P} + T_{\hat{P}/P}\left[\left(\omega_{P/C}\right)_P^X\right] \\
&= \left[T_{\hat{P}/P}\left[\left(\omega_{P/C}\right)_P^X\right] T_{P/\hat{P}} - \left[\left(\hat{\omega}_{P/C}\right)_P^X\right]\right] T_{\hat{P}/P}
\end{aligned}$$

or finally:

$$\dot{T}_{\hat{P}/P} = \left[\left\{T_{\hat{P}/P}\left(\omega_{P/C}\right)_P - \left(\hat{\omega}_{P/C}\right)_P\right\}^X\right] T_{\hat{P}/P} \tag{85}$$

To continue, denote:

$$Q = T_{\hat{P}/P} - I \tag{86}$$

So that equation (83) becomes:

$$\delta T_{C/P} = -T_{C/P}(I+Q)^{-1} Q \tag{87}$$

and equation (85) becomes:

$$\dot{Q} = \left[\left\{(I+Q)\left(\omega_{P/C}\right)_P - \left(\hat{\omega}_{P/C}\right)_P\right\}^X\right] (I+Q) \tag{88}$$

But:

$$\left(\omega_{P/C}\right)_P = \left(\omega_{P/I}\right)_P - \left(\omega_{E/I}\right)_P \tag{89}$$

and, using equations (69), (70), and (76):

$$\left(\hat{\omega}_{P/C}\right)_P = \left(\omega_{P/I}\right)_P - (I+Q)\left(\omega_{E/I}\right)_P - \delta\Delta\omega \tag{90}$$

for both the strapdown and rotationally free platform cases, where:

$$\delta\Delta\omega = \hat{\Delta\omega} - \Delta\omega \tag{91}$$

Using equations (89) and (90) in (88) leads to:

$$\boxed{\dot{Q} = \left[\left\{Q\left(\omega_{P/I}\right)_P + \delta\Delta\omega\right\} X\right] (I+Q)} \tag{92}$$

Equations (77) through (81), (87), and (92) together comprise a set of non-linear error equations in the closed-loop error variables δV, δP or δp, and Q, and in the forcing error variables δf_p and $\delta\Delta\omega$. These equations must however, be linearized for use in designing the processor Kalman filter. There are two principal steps involved in doing this: a) the linearization of the dynamic platform-to-computer misalignment error [equation (92)], and the linearization of the gravity error [equation (81)].

To linearize equation (92), denote by ψ the vector representing the instantaneous angular misalignment between the estimated and actual attitude of the inertial platform. Using equation (38) of Appendix I, it follows that:

$$T_{P/\hat{P}} = I \cos|\psi| + (1-\cos|\psi|)\, uu^T - \sin|\psi|\,(ux) \tag{93}$$

where $u = \psi/|\psi|$ is the unit vector about which the rotation angle $|\psi|$ is defined.

If $|\psi|$ is small, $T_{P/\hat{P}}$ can therefore be linearly approximated by:

$$T_{P/\hat{P}} \approx I - (\psi X) \tag{94}$$

So that:

$$Q = (\psi X) \tag{95}$$

Using this result in (92) gives:

$$(\dot{\psi}_P X) \approx \left[\left\{ (\psi_P X) \ (\omega_{P/I})_P + \delta \Delta \omega \right\} X\right] \left[I + (\psi_P X)\right] \tag{96}$$

So that, to the first order:

$$\boxed{\dot{\psi}_P = - \left[(\omega_{P/I})_P X\right] \psi_P + \delta\Delta\omega} \tag{97}$$

Equation (81) can be linearized as follows. To start, note that:

$$g(P) = G(P) + \Delta g(P) \tag{98}$$

where $\Delta g(P)$ is the small difference between mass attraction and plumb-bob gravity. Since $|\Delta g|$ is at most less then 1/3 of one percent of $|g|$, it can be neglected in deriving a linearized error relationship between δg and δP.

Thus:

$$\delta g = \hat{g}(\hat{P}) - g(P) \approx \delta g_A + G(\hat{P}) - G(P)$$

or

$$\delta g = \delta g_A - C_G \delta P \tag{99}$$

where δg_A is the error in gravity due to sources other than position error (e.g., gravity anomalies) and C_G is the matrix relation between the position-error-dependent gravity error, and the position error. In particular:

$$C_G \approx -\frac{\partial G}{\partial P} \tag{100}$$

For navigation in the E frame, C_G can be shown to be given by*.

$$(C_G)_E = \frac{|G_E(P_E)|}{|P_E|}\left[I - 3\left(\frac{g_E}{|g_E|}\right)\left(\frac{g_E}{|g_E|}\right)^T\right] \tag{101}$$

On the other hand, for navigation in the EF frame, δg becomes:

$$\delta g_{EF} = T_{EF/E}\,\delta g_A - T_{EF/E}(C_G)_E\,\delta P_E \tag{102}$$

But since $\delta P_E = T_{E/EF}\,\delta p_{EF}$, it follows that:

$$\delta g_{EF} = T_{EF/E}\,\delta g_A - (C_G)_{EF}\,\delta p_{EF} \tag{103}$$

where

$$(C_G)_{EF} = T_{EF/E}\,(C_G)_E\,T_{E/EF} \tag{104}$$

Substitution of equation (98) in (101) leads to:

$$(C_G)_{EF} = \frac{|G_{EF}(P_{EF})|}{|P_{EF}|}\left[I - 3\left(\frac{g_{EF}}{|g_{EF}|}\right)\left(\frac{g_{EF}}{|g_{EF}|}\right)^T\right] \tag{105}$$

Thus C_G can be represented in either the E or EF frames by:

$$\boxed{C_G = \frac{|G(P)|}{|P|}\left[I - 3\left(\frac{g_{EF}}{|g_{EF}|}\right)\left(\frac{g_{EF}}{|g_{EF}|}\right)^T\right]} \tag{106}$$

and the gravity error by:

$$\boxed{\delta g = \delta g_A - C_G\,\delta P \text{ (OR}\,\delta p)} \tag{107}$$

where δg_A is the non-position-error-dependent gravity error expressed in the C frame.

---

* See G. Pitman (ed.),"Inertial Guidance", J. Wiley & Sons, 1962, Chapter 1.

Finally, substitution of equation (95) in (87), and substitution of the result in (78) leads, after linearization, to:

$$\boxed{\delta f_C = T_{C/P}\left\{\delta f_P + (f_P X)\ \psi_P\right\}} \qquad (108)$$

and substitution of equation (107) into (79) leads to:

$$\boxed{\delta \dot{V} = \delta f_C + \delta g_A - C_G\ \delta P\ (\text{or } \delta p) - 2\ (\omega_{E/I} X)\ \delta V} \qquad (109)$$

Equations (108), (109), (80), and (97) together comprise the desired set of linearized inertial navigation error equations.

Switching of the error equations must be carried out when the navigation equations are switched. The appropriate error switching equations are obtained as follows.

Direct linear perturbation of equations (61 a,b) and (64 a,b) gives*:

$$\begin{array}{ll} \text{a)}\ \delta v_{EF} = T_{EF/E}\ \delta v_E & \\ & (\text{E to EF}) \\ \text{b)}\ \delta p_{EF} = T_{EF/E}\ \delta p_E & \end{array} \qquad (110)$$

$$\begin{array}{ll} \text{a)}\ \delta v_E = T^T_{EF/E}\delta v_{EF} & \\ & (\text{EF to E}) \\ \text{b)}\ \delta p_E = T^T_{EF/E}\delta p_{EF} & \end{array} \qquad (111)$$

Also, since the location of the EF frame is known with respect to the E frame (see footnotes) and the earth rate vector is known in the E frame, this vector is also known errorlessly in the EF frame. For this same reason, no change in the platform-to-computer misalignment ψ_P is required

---

* C_E and $T_{EF/E}$ are errorless by definition; i.e., the location of the center of the EF frame in, and its attitude with respect to, the E frame are assumed known, although the position and velocity of users and/or emitters and/or other points in the EF frame with respect to the EF frame are in general assumed to be in error. This philosophy leads to a more compact overall mechanization than do alternative philosophies.

at the time of C frame switching, since the C frame is known, be it E or EF. No additional error contribution to ψ_P is therefore introduced by switching.

Equations (110) and (111) therefore define all necessary error switching operations.

APPENDIX IV

ADR NAVIGATION EQUATIONS

This appendix derives the processor ADR equations, predicated on the frame selection C=EF.

In ADR, since the basic driving DR navigation information is available at the velocity, rather than the acceleration level, no acceleration-level processing is required. Integral processing is therefore confined to the equation:

$$\dot{p} = v \tag{112}$$

The velocity v for this equation is obtained from:

$$v = v_W + v_{AS} \tag{113}$$

The C-frame-referenced airspeed vector v_{AS} is obtained from:

$$v_{AS} = T^T_{A/C}\left(v_{AS}\right)_A \tag{114}$$

and the airframe (A frame) referenced airspeed vector from:

$$\left(v_{AS}\right)_A = T_{A/A'}k_{A/A'}\left(\left|v_{ASM}\right| + \left|\Delta v_{ASM}\right|\right) \tag{115}$$

where $\left|v_{ASM}\right|$ is the measured (scalar) airspeed, $\left|\Delta v_{ASM}\right|$ is the airspeed bias and scale factor correction, and $T_{A/A'}k_{A/A'}$ is a combined angle of attack correction and vectorizing operator (i.e., to convert airspeed from a scalar to a vector). Finally the A-to-C-frame transformation required by equation (114) is generated from:

$$T_{A/C} = T_{A/P}T_{P/C} \tag{116}$$

and $T_{A/P}$ from:

$T_{A/P}$ = FUNCTION OF AHRU ATTITUDE READOUTS

Discussion of $T_{P/C}$ generation is deferred for the moment since it relates to wind vector generation for equation (113).

Equation (113) requires a C-frame-referenced wind vector. In ADR, generation of wind in any frame is of course based on some initial, a priori wind vector estimate which is subsequently decayed in the absence of further information.* A natural frame in which to conduct this basic wind estimation process is the L (locally level) frame. In particular, to maintain maximum inter-DR-mode commonality of processor algorithms, this L frame should be the same as that used in IDR; i.e., a wander azimuth frame. The wind processing equations are therefore:

$$v_W = T^T_{L/C}\left(v_W\right)_L \tag{117}$$

$$v_{W\,L} = Q_{WL}\left(v_W\right)_L \tag{118}$$

where Q_{WL} is an appropriate, wind estimate decay (diagonal) matrix.

Returning now to consideration of $T_{P/C}$ generation for equation (116), a natural way, which makes use of $T_{L/C}$, is:

$$T_{P/C} = T_{P/L}T_{L/C} \tag{119}$$

This equation of course requires separate generation of $T_{P/L}$ and $T_{L/C}$ by means of

$$\dot{T}_{L/C} = \left(\omega_{L/C}\right)_L \times T_{L/C} \tag{120}$$

$$\dot{T}_{P/L} = \left(\omega_{P/L}\right)_P \times T_{P/L} \tag{121}$$

where $(\omega_{L/C})_L$ is obtained (as in IDR for P = L) from:

$$\left(\omega_{L/C}\right)_L = K_L T_{L/C} v \tag{122}$$

and $\left(\omega_{P/L}\right)_P$ from:

*This estimate can of course be corrected (e.g., via a Kalman filter) if external measurements of groundspeed (e.g., from continuous pseudoranging and/or pseudorange-rating) are available.

$\left(\omega_{P/L}\right)_P$ = 0 if AHRU platform is in DG(wander azimuth) mode

= local meridian convergence rate (i.e., azimuth rate equal to vehicle longitude rate multiplied by the sine of vehicle geographic latitude)

Finally, again to maintain maximum inter-DR-mode processor algorithm commonality, since (120) and (121) must be computed in ADR and strapdown IMU IDR anyway, then if the hardware configuration involves both an AHRU and a rotationally free IMU, (120) and (121) should also be implemented in IDR, instead of the simpler equation (44) of Appendix II alone. In addition, carrying the additional $T_{P/L}$ transformation desirably generalizes processor IMU acceleration data processing and torquing control rate capabilities.

APPENDIX V

PDR NAVIGATION AND ERROR EQUATIONS

This appendix discusses the PDR mechanization and error equations based on the assumption C=EF.

1. PDR NAVIGATION EQUATIONS

The C-Frame-referenced aircraft acceleration equation is:

$$\frac{dv}{dt} = f + g - 2\omega_{E/I} \times v \qquad (123)$$

where, as usual, the C-frame subscripting is omitted for brevity.

The corresponding equation referenced to an L frame is:

$$\frac{d_L v_L}{dt} = f_L + g_L - \left(2\omega_{E/I} + \omega_{L/C}\right)_L \times v_L \qquad (124)$$

and the two equations are related by:

$$\frac{dv}{dt} = T^T_{L/C}\left\{\frac{d_L v_L}{dt} + \left(\omega_{L/C}\right)_L \times v_L\right\} \qquad (125)$$

Now consider in particular the L frame defined by the orthogonal unit vectors*:

$$u_1 = \frac{g}{|g|},\ u_2 = \frac{\gamma \times u_1}{|\gamma \times u_1|},\ u_3 = u_1 \times u_2,\ \gamma = \frac{v}{|v|} \qquad (126)$$

i.e., the locally level frame with its horizontal axes down-and cross-ground-track. In particular, the $T_{L/C}$ transformation can be defined conveniently in terms of these unit basis vectors as:

$$T_{L/C} = \left[u_1 \vdots u_2 \vdots u_3\right]^T \qquad (127)$$

*Note that this L frame, although it has the same vertical axis as the wander azimuth L frame used in IDR and ADR, has its horizontal axes constrained to follow the horizontal projection of the vehicle velocity vector.

Consider the bracketed term on the right-hand side of (125). If the aircraft is flying at constant speed and altitude (this does not preclude turns) then $\frac{d_L v_L}{dt} = 0$, and the overall term reduces to:

$$(\omega_{L/C})_L \times v_L = \begin{bmatrix} \omega_{L/C1} \\ \omega_{L/C2} \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ v_{L3} \end{bmatrix} = \begin{bmatrix} \omega_{L/C2} v_{L3} \\ -\omega_{L/C1} v_{L3} \\ 0 \end{bmatrix} \tag{128}$$

Thus, if the aircraft is not turning, only the earth-centripetal (radial) term $\omega_{L/C2} v_{L3}$ is present. If the aircraft is also turning, then the additional turn-centripetal term $-\omega_{L/C1} v_{L3}$ is required as well.

If, further, the aircraft is either or both horizontally (along-track) and vertically accelerating or decelerating, then:

$$\frac{d_L v_L}{dt} = \begin{bmatrix} \dot{v}_{L1} \\ 0 \\ \dot{v}_{L3} \end{bmatrix} \tag{129}$$

combining results:

$$\frac{d_L v_L}{dt} + (\omega_{L/C})_L \times v_L \approx \begin{bmatrix} -\frac{v_{L3}^2}{|P_E|} \text{ (always)} + \dot{v}_{L1} \text{ (start or end of climb)} \\ -\omega_{L/C1} v_{L3} \text{ (in turn)} \\ \dot{v}_{L3} \text{ (horizontal acceleration/deceleration)} \end{bmatrix} \tag{130}$$

In PDK, the horizontal (along-track) and vertical accelerations $\dot{v}_{L3}$ and $\dot{v}_{L1}$, and the turning rate (necessary to determine the cross-track turning acceleration) are not directly measurable, but must be inferred (after integration, using pseudorange and/or range-rating data). Calling the acceleration vector composed of these three components β, then the overall, C-Frame-referenced acceleration can be written:

$$\dot{v} = T_{L/C}^{T}\left(\beta_L - a_L\right) = \beta - a \qquad (131)$$

where

$$\beta_L = \begin{bmatrix} \beta_{L1} \\ \beta_{L2} \\ \beta_{L3} \end{bmatrix} \qquad a_L = \begin{bmatrix} -\left\{|v|^2 - \left(v^T u_1\right)^2\right\} \\ 0 \\ 0 \end{bmatrix}$$

The β_L components can be simply modeled for use in this equation as exponentially time-correlated variables; i.e.:

$$\dot{\beta}_{Li} = -\frac{1}{\tau_{\beta Li}}\beta_{Li} \quad (i = 1,2,3) \qquad (132)$$

where the correlation times $\tau_{\beta Li}$ depend on the direction u_i involved, and on the type of maneuver, per the following table:

	No Maneuvers	Start of Climb or Descent	Along-Track Acceleration or Deceleration	Turn
$\tau_{\beta L1}$	$\tau_{\beta LN1}$	$\tau_{\beta LS1}$	$\tau_{\beta LN1}$	$\tau_{\beta LN1}$
$\tau_{\beta L2}$	$\tau_{\beta LN2}$	$\tau_{\beta LN2}$	$\tau_{\beta LN2}$	$\tau_{\beta LS2}$
$\tau_{\beta L3}$	$\tau_{\beta LN3}$	$\tau_{\beta LN3}$	$\tau_{\beta LS3}$	$\tau_{\beta LS3}$

$\tau_{\beta LNi}$ = Nominal, no-maneuver value

$\tau_{\beta LSi}$ = Shorter, maneuver value

In particular, the shorter correlation times are used for both components of horizontal acceleration throughout a turn to account for the change in effective acceleration due to the change in the groundtrack direction relative to wind direction.

Information from the flight control system (e.g., throttle setting, control surface settings, etc.) could presumably be used to control the maneuver indicators required by implication in the above $\tau_{\beta Li}$-setting table. If not, single, fixed values (which would have to be intermediate, compromise values between the no-maneuver and maneuver values) would have to be used instead.

Combining results, the PDR mechanization equations are:

$$\dot{v} = T^T_{L/C}(\beta_L - a_L) \qquad a_L = \left\{|v|^2 - \left(v^T u_1\right)^2\right\}(u_1)_L \tag{133}$$

$$T_{L/C} = \left[u_1 \vdots u_2 \vdots u_3\right]^T \tag{134}$$

$$u_1 = \frac{g}{|g|},\ u_2 = \frac{\gamma x u_1}{\gamma x u_1},\ u_3 = u_1 x u_2,\ \gamma = \frac{v}{|v|} \tag{135}$$

$$\dot{\beta}_L = -K_{\beta L}\beta_L \qquad \beta_L = \begin{bmatrix} \beta_{L1} \\ \beta_{L2} \\ \beta_{L3} \end{bmatrix} \qquad K_{\beta L} = \begin{bmatrix} k_{\beta L1} & 0 & 0 \\ 0 & k_{\beta_{L2}} & 0 \\ 0 & 0 & k_{\beta L3} \end{bmatrix} \tag{136}$$

$$k_{\beta Li} = \frac{1}{\tau_{\beta Li}} \quad (i = 1,2,3)$$

$\tau_{\beta i}$ settings per table from flight control data, or single, fixed values

(p, p_E, g_E, g, $|h|$ equations same as for IDR)

2. PDR ERROR EQUATIONS

In the acceleration equation (131),

$$\dot{v} = T^T_{L/C}(\beta_L - a_L)$$

the terms β_L and a_L model L frame accelerations, and $T^T_{L/C}$ is the matrix necessary to express them in the C frame. By direct linear perturbation, the corresponding error equation is:

$$\delta\dot{v} = T^T_{L/C}\delta\beta_L + \delta T^T_{L/C}(\beta_L - a_L) - T^T_{L/C}\,\delta a_L \tag{137}$$

But since both $T^T_{L/C}$ and a_L are just velocity dependent, while only β_L is acceleration dependent, for short-time extrapolations the last two terms can be neglected so that:

$$\delta\dot{v} \approx T^T_{L/C}\delta\beta_L \tag{138}$$

Also, by direct perturbation:

$$\begin{cases} \delta\dot{\beta}_L = -K_{\beta L}\delta\beta_L \\ \delta\dot{p} = \delta v \end{cases} \tag{139}$$

No gravity error feedback (via position error) coupling into the acceleration error equation is needed here, since computed gravity is not used in the acceleration equation (except to compute the direction u_1; but $\delta T^T_{L/C}$ error effects are neglected).

APPENDIX VI

BASIC MEASUREMENT MATRICES DERIVATION

This appendix is devoted to the derivation of the basic measurement (M) matrices for the following general types of measurements: range, range-rate and barometric altitude. A high level of commonality between the various M matrices was considered to be of primary importance in the development.

In taking measurements (range or range-rate) relative to emitters the question arises as to whether or not emitter, as well as user location errors should be modeled and estimated. If it is decided that emitter location errors should be estimated, then there is the possibility these errors are interrelated. If this is so, and the interrelationships are known, then the number of error state variables required to be carried by the Kalman filter may be significantly reduced. The number of measurements required to estimate the emitter errors would also be reduced by a like amount.

Consider, for example, a situation in which the user is operating within range of a network consisting of four, earth fixed, range type emitters. Assume that emitter position errors are to be estimated. Now if the emitter location errors are not correlated, or if they are but this fact is not used in the estimation process, then a total of twelve state error variables are required in the filter model. Further, twelve different scalar range measurements are required to estimate the twelve state error variables.

Depending upon the circumstances, a second alternative might exist. Consider the case where the first emitter is located by some means and then the second, third and fourth emitters in the network are located relative to the first. Assume that distances between emitters is known quite precisely but that errors in local vertical and true north are present, in the process of locating these emitters. In this situation, the four emitters might be considered as four points in a rigid body which has three translational error components (those due to the location errors of the first emitter) and three rotational error components (those due to errors in local vertical and true north). For this case, the position errors of all four emitters may be estimated using only six error variables and six different scalar range measurements. If, during the locating of the last three emitters, true north and local vertical were also known precisely, then only three error variables are required. Finally, under certain circumstances, it may be that no emitter error state variables are required, i.e., emitter locations are known precisely and no emitter position error estimates are required.

In the derivations of this appendix, the worst case has been assumed, i.e., the M matrices provide for the inclusion of three error variables for each emitter whose position error is to be estimated.

Range measurements are of two general types, line-of-sight (LOS) range measurements and earth-mode (EM) range measurements. Earth-mode measurements (e.g., LORAN or Omega) are processed differently than are LOS measurements. But more importantly, from the standpoint of generating M matrices, the information content in an EM range measurement is less. Since EM signal propagation is based upon the concept of a spherical waveguide, there is no way to relate changes in user altitude to changes in measured range. For this reason EM range measurements must be restricted such that they are used only to estimate position errors in the local horizontal plane (local horizontal at the user and emitter positions); LOS and EM measurements are therefore discussed below in separate subsections.

There are a number of symbols which are used in this appendix and which are not defined in the glossary. Some are defined in the text at the time they are used; however, some which are used more generally throughout this appendix are defined below:

Symbol Definitions

[--- UNITY ---] is a row vector in which every element is unity

[--- ZERO ---] is a row vector in which every element is zero

Superscript T = Transpose of the quantity

$\delta R'_{Tj} = -\delta R_{Tj}$ = the sum of all scalar time correlated errors associated with a specific j^{th} radio ranging measurement

$\delta \dot{R}'_{Tj} = -\delta \dot{R}_{Tj}$ = the sum of all scalar time correlated errors associated with a specific j^{th} radio range-rate measurement

$\frac{\partial F}{\partial P} \triangleq \left[\frac{\partial F}{\partial X}\ \frac{\partial F}{\partial Y}\ \frac{\partial F}{\partial Z}\right]$ = gradient of F

The gradient of scalar functions is used throughout this appendix (both implicitly and explicity) to expand computed functions about their actual values using a Taylor series expansion. Second and higher order terms in these expansions are always neglected (to arrive at linear relationships), which is valid when the computed value is very close to the actual value. This condition is always assumed. Further, since the actual values of the function argument variables are never known, the resulting partial derivatives are evaluated using the estimated values of these variables. This is also reasonable where the errors are very small.

In an attempt to avoid derivations within a section which detract from the main thought, three identities (which are used within the sections) are defined and proved as follows.

Given a vector b,

$$\delta|b| = \frac{b^T}{|b|}\delta b, \quad \text{where } \frac{b}{|b|} \text{ is a unit vector pointing in the direction of b.} \tag{140}$$

$$\frac{d}{dt}|b| = \frac{b^T}{|b|}\frac{d(b)}{dt} \tag{141}$$

$$\delta\left\{\frac{b}{|b|}\right\} = \left\{I - \frac{b\ b^T}{|b|\,|b|}\right\}\frac{\delta b}{|b|}\ ; \quad I \triangleq \text{Identity Matrix} \tag{142}$$

Proof of (140):

$|b|\,|b| = |b|^2 = b^Tb$, which implies that

$$\delta\left\{|b|\,|b|\right\} = \delta\left\{b^Tb\right\}$$

Evaluating both sides yields:

$$2|b|\,\delta|b| = 2b^T\delta b \Rightarrow \delta|b| = \frac{b^T}{|b|}\delta b$$

Proof of (141):

$$\frac{d}{dt}|b|^2 = \frac{d}{dt}(b^Tb)$$

Again evaluating both sides,

$$|b|\frac{d}{dt}|b| = b^T\frac{d(b)}{dt} \Rightarrow \frac{d|b|}{dt} = \frac{b^T}{|b|}\frac{d}{dt}(b)$$

Proof of (142):

$$\delta\left\{\frac{b}{|b|}\right\} = \frac{|b|\,\delta(b) - (b)\,\delta|b|}{|b|\,|b|}$$

Substituting in the results of (140) above yields

$$\delta\left\{\frac{b}{|b|}\right\} = \frac{|b|\ \delta(b)}{|b|\ \ |b|} - \left(\frac{b\ b^T}{|b|\,|b|}\right)\frac{\delta b}{|b|} = \left\{I - \frac{b\ b^T}{|b|\,|b|}\right\}\frac{\delta b}{|b|}$$

1. RANGE MEASUREMENTS (LINE-OF-SIGHT)

In this subsection, M matrices are developed for line-of-sight range measurements which are consistent with both the C=E and C=EF computational frames. For the C=E frame case, the range between the user and the j^{th} emitter is given by

$$\left| R_j \right| = \left| P + d - E_j \right| \tag{143}$$

where P, d and E are as defined in the symbol glossary. Briefly, P is the vector distance from the center of the earth to the center of the user platform frame; d is the vector distance from the center of the user platform frame to the user receiving antenna; E_j is the vector distance from the center of the earth to the j^{th} emitter antenna.

Now, let

$$\left| R_{mj} \right| = \left| R_j \right| + \delta R'_{Tj} + n_{Rj}(t) \tag{144}$$

and

$$\left| \hat{R}_j \right| = \left| R_j \right| + \delta \left| R_j \right| \tag{145}$$

represent, respectively, the measured and computed values corresponding to the range between the user and the j^{th} emitter.

To generate the M_j measurement matrix, the measured range is subtracted from the computed range after the $\delta|R_j|$ term in equation (145) has been expanded in terms of system variables. Taking the first variation of $|R_j|$, defined in equation (143), yields:

$$\delta \left| R_j \right| = \delta \left| P + d - E_j \right| \tag{146}$$

Now, using equation (140), (identity #1),

$$\delta \left| P + d - E_j \right| = \frac{(P + d - E_j)^T}{\left| P + d - E_j \right|} \cdot (\delta P + \delta d - \delta E_j) \tag{147}$$

where the vector quantity,

$$\frac{(P + d - E_j)}{\left| P + d - E_j \right|}$$

is, by definition, a unit vector which points from the j^{th} emitter antenna to the user antenna. Let this unit vector be defined as r_j. Using this definition, one can rewrite equation (147) as

$$\delta|P+d-E_j| = r_j^T \cdot (\delta P+\delta d - \delta E_j) \tag{148}$$

Now, subtracting equation (144) from (145) and utilizing the results of equations (146) and (148) produces the following equation:

$$|\Delta R_j| = r_j^T \cdot (\delta P + \delta d - \delta E_j) + \delta R_{Tj} + n_{Rj}(t) \tag{149}$$

It turns out, as will be shown later, that the term δd in equation (149) provides a means of estimating errors in the transformation matrix, $T_{C/A}$, which relates the computation frame and the aircraft frame. Neglecting for the moment the δd term, equation (149) in vector-matrix notation becomes:

$$|\Delta R_j| = \left[r_j^T \; \vdots \; [ZERO] \; \vdots \; -r_j^T \; \vdots \; [ZERO] \; \vdots \; [UNITY] \; \vdots \; [ZERO] \right] \begin{bmatrix} \delta P \\ - - - \\ \delta E_j \\ - - - \\ \delta R_{Tj} \\ - - - \end{bmatrix} + n_{Rj}(t) \tag{150}$$

From equation (150), the M_j matrix is, by definition,

$$M_j = \left[r_j^T \; \vdots \; [ZERO] \; \vdots \; -r_j^T \; \vdots \; [ZERO] \; \vdots \; [UNITY] \; \vdots \; [ZERO] \right] \tag{151}$$

In equation (149), the vector d represents the vector distance from the center of the platform frame to the user antenna expressed in C frame coordinates. If the effect of this separation distance is to be taken into account (both for range and range-rate measurements) then the vector d must be an input parameter. However, since d is physically measured in airframe coordinates, then the input will be in airframe coordinates. The computational algorithms then must take into account the necessary conversions.

Let d_A represent the physical measurement of d expressed in airframe coordinates; then

$$d = T_{C/A} \; d_A \tag{152}$$

Using equation (152),

$$\delta d = \delta Tc_{/A}\, d_A + Tc_{/A}\, \delta d_A \tag{153}$$

but $\delta d_A = 0$ since d_A is a known constant vector in airframe coordinates; hence, equation (153) reduces to

$$\delta d = \delta Tc_{/A} d_A \tag{154}$$

Now, let the 3x3 error matrix, $\delta Tc_{/A}$, be represented by its three-row vector partitions; i.e., let

$$\delta Tc_{/A} = \begin{bmatrix} [\delta T_1] \\ [\delta T_2] \\ [\delta T_3] \end{bmatrix} \tag{155}$$

Using equation (16), one can now rewrite (154) as:

$$\delta Tc_{/A} d_A = \begin{bmatrix} \delta T_1 \bullet d_A \\ \delta T_2 \bullet d_A \\ \delta T_3 \bullet d_A \end{bmatrix} = \begin{bmatrix} d_A^T \bullet \delta T_1^T \\ d_A^T \bullet \delta T_2^T \\ d_A^T \bullet \delta T_3^T \end{bmatrix} = \begin{array}{cc} (3x9) & (9x1) \\ \left[\begin{array}{c|c|c} d_A^T & 0 & 0 \\ \hline 0 & d_A^T & 0 \\ \hline 0 & 0 & d_A^T \end{array}\right] & \left[\begin{array}{c} \delta T_1^T \\ \hline \delta T_2^T \\ \hline \delta T_3^T \end{array}\right] \end{array} \tag{156}$$

To simplify notation, let

$$\delta Tc_{/A} d_A = D \delta Tc'_{/A} \tag{157}$$

where D is understood to be the 3x9 matrix in equation (156) and $\delta T'_{c/A}$ is the (9x1) vector representation of the error matrix $\delta Tc_{/A}$. The $r_j^T \bullet \delta d$ term in equation (149) now becomes

$$r_j^T \cdot \delta d = \left[r_j^T D\right] \delta Tc'_{/A} \tag{158}$$

where $\left[r_j^T D\right]$ is a (1x9) row vector.

Now, if errors in the $Tc_{/A}$ transformation matrix are to be estimated, then the M_j matrix and state error vector, defined by equation (150), are modified and become:

$$\left|\Delta R_j\right| = \left[\, r_j^T \;\vdots\; [r_j^T D] \;\vdots\; [ZERO] \;\vdots\; -r_j^T [ZERO] \;\vdots\; [UNITY] \;\vdots\; [ZERO] \,\right] \begin{bmatrix} \delta P_T \\ \delta Tc_{/A} \\ \vdots \\ \delta E_j \\ \vdots \\ \delta R_{Tj} \end{bmatrix} + n_{Rj}(t) \tag{159}$$

In the C = EF frame case, the center of the EF frame is translated from the center of the E frame through the vector distance C; and the coordinate axes of the EF frame are related to the E frame coordinate axes through the orthogonal transformation $T_{EF/E}$.

To fix ideas in the following derivation, use is made of the following vector diagram:

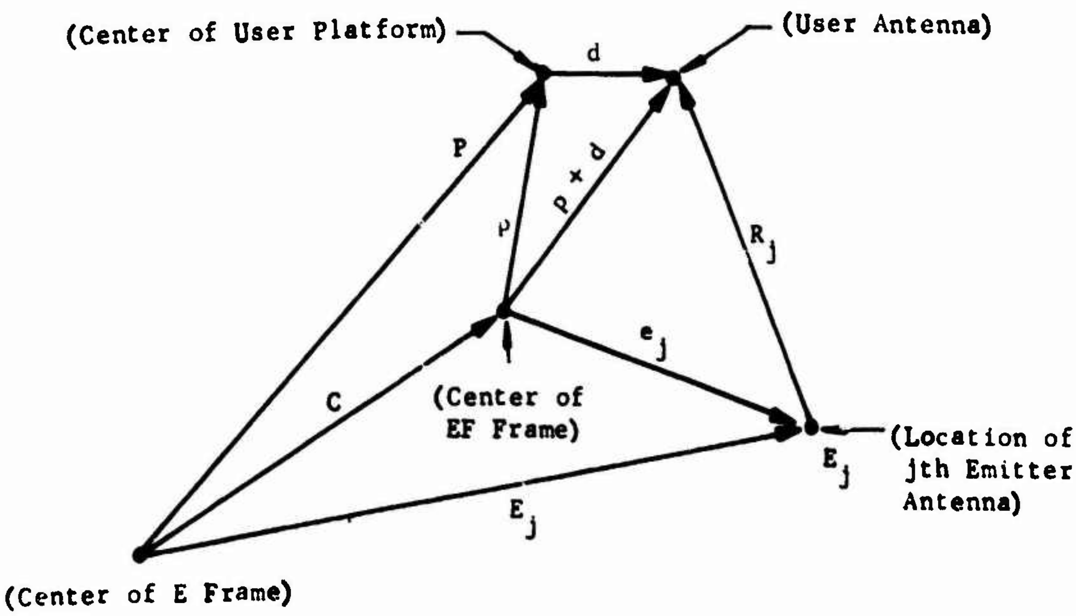


Figure 33. LOS Emitter/User Geometry

where, now p+d is the vector distance of the user antenna with respect to the EF frame center, and e_j is the vector distance of the jth emitter antenna with respect to the center of the EF frame.

Now, from Figure ,

$$\left|R_j\right| = \left|P + d - E_j\right| = \left|p + d - e_j\right| \tag{160}$$

The measured range is (as before) defined by equation (144). Now, let the computed range be defined by:

$$\left|\hat{R}_j\right| = \left|R_j\right| + \delta\left|R_j\right| = \left|p + d - e_j\right| + \delta\left|p + d - e_j\right| \tag{161}$$

Following the identical procedure as described above to evaluate the $\delta\left|p + d - e_j\right|$ term in equation (161) yields

$$\delta\left|p + d - e_j\right| = r_j^T \cdot (\delta p + \delta d - \delta e_j) \tag{162}$$

where r_j^T; δp, δd and δe_j are all expressed in C = EF frame coordinates. The δd term can, as before, be used to estimate errors in the transformation matrix which relates the C = EF frame and the user aircraft frame. (Note: In the previous consideration, $T_{A/c} = T_{A/E}$, whereas here $T_{A/c} = T_{A/EF}$.)

As a result, equation (162) yields an M_j matrix which is identical to the M_j matrix defined by equation (150) (if the δd term is neglected), or equation (159) (if the δd term is used to estimate transformation errors). The M matrix elements are computed in the operating frame coordinates, using parameters measured relative to the center of this frame.

2. RANGE MEASUREMENTS (EARTH MODE)

As described above, the information contained in an earth mode range measurement must be restricted such that it is used only to estimate position errors in the local horizontal plane, at both the user and the emitter locations. Even with this restriction it is still possible to derive M matrices for this mode of operation which display a certain level of commonality with the M matrices developed for LOS measurements. This commonality is achieved using the same unit vector, r_j, developed in Section 1 above, and then rotating this unit vector into the local horizontal plane. The following paragraphs describe this procedure in detail, in which the C = E frame case is considered first.

To provide a good insight into the derivations of this section, use is made of the following sketch where for simplicity the vector A is used ($A = P + d$) rather than $P + d$; "user position" implies user antenna position and "emitter location" implies emitter antenna location.

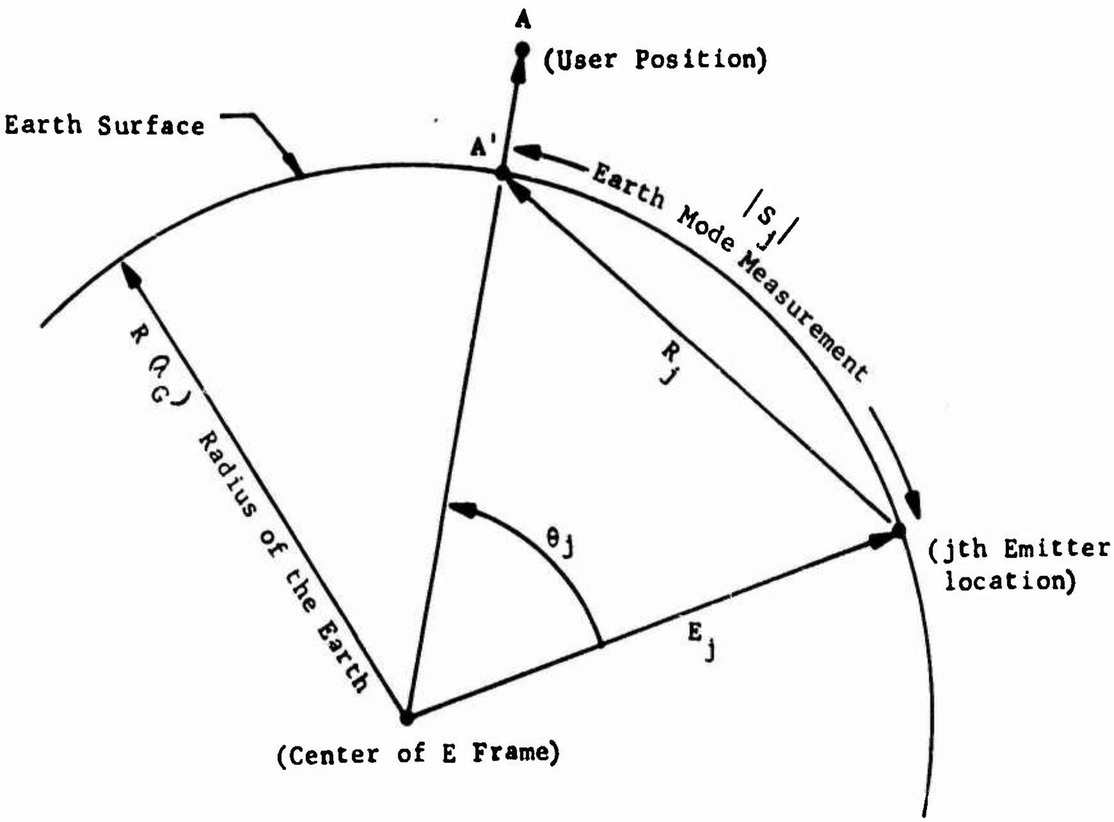


Figure 34. Earth-Mode Emitter/User Geometry (1)

In this sketch, A' is the point on the earth's surface directly below the user position, λ_G is the geocentric latitude and θ_j is the angle between the vectors A' and E_j. Let $|S_j|$ represent the actual distance between the emitter position and the point A' as measured along the surface of the earth.

The computed value of the distance $|S_j|$ can then be expressed by the following line integral:

$$|\hat{S}_j| = \int_{\hat{E}_j}^{\hat{A}'} |\hat{R}(\lambda_G)| \, d\hat{\theta}_j \qquad (163)$$

and the measured value of $|S_j|$ can be expressed as

$$|S_{mj}| = |S_j| + \delta R'_{\tau j} + n_{Rj}(t) \qquad (164)$$

The value $|\Delta S_j|$ then is arrived at by subtracting equation (164) from (163). The M matrices for this mode of operation are not, however, developed by operating on equation (163) because of the altitude restriction, but rather as follows.

As noted in Section 1 above, r_j is a unit vector which points from the emitter to the user along the lineof sight. Now, assume that the line of sight distance between the emitter and user ($|R_j|$, where R_j is as described in the sketch) is much greater than the altitude of the user. Under this assumption, and for the purposes of the following derivation, it is reasonable to assume that the user is located at the point A'. (If this assumption is not valid, then user altitude must be subtracted out of the vector A to arrive at A'. The derivation then continues as described below.

The following sketch describes how the earth mode M matrix is derived using the results of the preceding section.

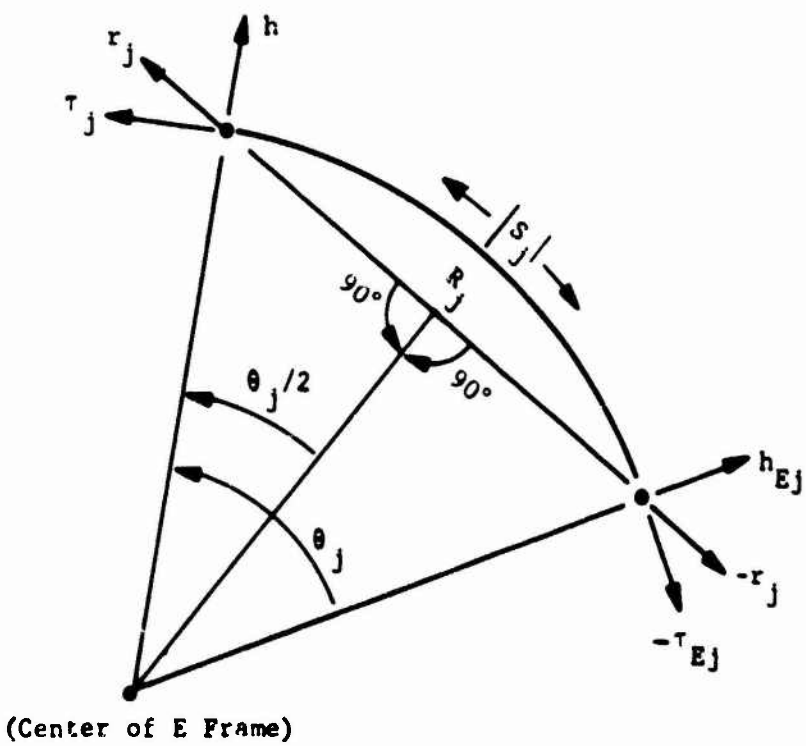


Figure 35. Earth-Mode Emitter/User Geometry (2)

In Figure 35,

h = unit vector along local vertical at user position = $\frac{A}{|A|}$

$r_j = \frac{R_j}{|R_j|}$; as previously defined

τ_j = unit vector contained in both the local horizontal plane and the plane containing the center of the E frame, the jth emitter location and the user location

h_{Ej} = unit vector along local vertical at emitter position = $\frac{E_j}{|E_j|}$

τ_{Ej} = same description as τ_j except at jth emitter location.

It might be well to note that $\frac{A}{|A|}$ does not in fact point exactly along the local vertical. However it only differs from the local vertical by the eccentric anomaly which is, at worst, approximately 11 arc-minutes. For the purposes of the M matrix this is a very good approximation.

Now, from the sketch and the unit vector definitions

$$r_j = (\cos \theta_j/2)\, \tau_j + (\sin \theta_j/2)\, h \tag{165}$$

at the user location and

$$-r_j = -(\cos \theta_j/2)\, \tau_{Ej} + (\sin \theta_j/2)\, h_{Ej} \tag{166}$$

at the emitter location.

Solving equations (165) and (166) for τ_j and τ_{Ej} respectively yields

$$\left.\begin{aligned} \tau_j &= \frac{1}{\cos \theta_j/2}\left[r_j - (\sin \theta_j/2)\, h\right] \\ &\text{and} \\ -\tau_{Ej} &= \frac{1}{\cos \theta_j/2}\left[-r_j - \left(\sin \theta_j/2\right) h_{Ej}\right] \end{aligned}\right\} \tag{167}$$

The unit vectors τ_j and τ_{Ej} can now be used to restrict the earth mode measurement such that only local horizontal estimates of user and emitter position can be made, i.e., let

$$|\Delta S_j| = \tau_j^T \cdot \delta A - \tau_{Ej}^T \cdot \delta E_j + \delta R_{Tj} + n_{Rj}(t) \qquad (168)$$

where the radio navigation measurement errors have been added. Using equation (159), the $|\Delta S_j|$ equation for the jth emitter range measurement then becomes

$$|\Delta S_j| = \left[\tau_j^T \,\vdots\, [\text{ZERO}] \,\vdots\, -\tau_{Ej}^T \,\vdots\, [\text{ZERO}] \,\vdots\, [\text{UNITY}] \,\vdots\, [\text{ZERO}] \,\vdots\, \right] \begin{bmatrix} \delta A \\ \vdots \\ \delta E_j \\ \vdots \\ \delta R_{Tj} \end{bmatrix} + n_{Rj}(t) \qquad (169)$$

where $\delta A = \delta(P + d) = \delta P + \delta d$.

As for line of sight measurements, the δd term can be used to estimate errors in the transformation matrix, $T_{C/A}$, or it can be neglected. If it is neglected, then $\delta A = \delta P$. If it is not neglected, then a (1 x 9) row vector, $[\tau^T D]$, is added to the M matrix and a (9 x 1) error vector, $\delta T'_{C/A}$, as previously described in Section 1.

Now in comparing the form of the M_j matrix defined in equation (169) with, e.g., the LOS M_j matrix defined by equation (150) (for this comparison let $\delta A = \delta P$), it is seen that the difference between the two matrices can be identified as follows:

$$\tau_j^T = \left(\frac{1}{\cos \theta_j/2}\right) r_j^T - (\sin \theta_j/2)\, h^T$$

and (170)

$$-\tau_{Ej}^T = -\left(\frac{1}{\cos \theta_j/2}\right) r_j^T - (\sin \theta_j/2)\, h_{Ej}^T$$

For $\theta_j = 0$, equation (169) reduces to equation (150), as can be seen by inspection using equation (170). The same applies for the case where errors in $T_{A/C}$ are to be estimated. (For particularly high user altitude cases it may be desirable to scale τ_j by the factor $\frac{A'}{|A|}$ or $\frac{|A| - |h|}{|A|}$.)

An identical relationship exists between the M matrices for the case C = E frame, and the case C = EF frame except that, when operating in the EF frame:

$$h = \frac{C + p}{|C + p|}$$
$$h_{Ej} = \frac{C + ej}{|C + ej|} \tag{171}$$

where C (expressed in EF frame coordinates) is the vector joining the centers of the E and EF frames, ej and p are the jth emitter and user locations respectively as measured in the EF frame.

3. RANGE-RATE MEASUREMENTS

In this section, range-rate measurements are assumed to be made along the line-of-sight between the user antenna and the jth emitter antenna; i.e., $|\dot{R}_j|$ is defined as follows:

$$|\dot{R}_j| = \frac{d\,|R_j|}{dt} = \frac{d}{dt}\,|P + d - Ej| \tag{172}$$

From equation (141):

$$\frac{d}{dt}\,|P + d - Ej| = \frac{(P + d - E_j)^T}{|P + d - E_j|} \cdot (V + \dot{d} - V_{Ej}) \tag{173}$$

Or, using the unit vector, r_j, (172) can be rewritten as

$$|\dot{R}_j| = r_j^T \cdot (V + \dot{d} - V_{Ej}) \tag{174}$$

Now let $|\hat{\dot{R}}_j|$, the computed value of the range-rate, be expressed as

$$|\hat{\dot{R}}_j| = |\dot{R}_j| + \delta|\dot{R}_j| \tag{175}$$

and let $|\dot{R}_{mj}|$, the measured value of the range-rate, be expressed as

$$|\dot{R}_{mj}| = |\dot{R}_j| + \delta\dot{R}_T{}'_j + n_{\dot{R}j}(t) \tag{176}$$

Subtracting equation (176) from equation (175) yields:

$$|\Delta\dot{R}_j| = \delta|\dot{R}_j| + \delta\dot{R}_{Tj} + n_{\dot{R}j}(t) \tag{177}$$

Now, using equation (174), one can rewrite the $\delta|\dot{R}_j|$ term in equation (177) as

$$\delta|\dot{R}_j| = r_j^T \cdot (\delta V + \delta\dot{d} - \delta V_{Ej}) + (V + \dot{d} - V_{Ej}) \cdot \delta r_j \tag{178}$$

To evalute δr_j in equation (178), recall that

$$r_j = \frac{(P + d - E_j)}{|P + d - E_j|} = \frac{R_j}{|R_j|} \tag{179}$$

Then, using equation (142),

$$\delta r_j = \delta\left\{\frac{(P + d - E_j)}{|P + d - E_j|}\right\} = \left|I - r_j\, r_j^T\right|\left\{\frac{(\delta P + \delta d - \delta E_j)}{|P + d - E_j|}\right\} \tag{180}$$

Now, if the results of equations (177), (178), (179), and (180) are gathered together, $|\Delta\dot{R}_j|$ now becomes

$$|\Delta\dot{R}_j| = m_j^T(\delta P + \delta d - \delta E_j) + r_j^T(\delta V + \delta\dot{d} - \delta V_{Ej}) + \delta\dot{R}_{Tj} + n_{\dot{R}j}(t) \tag{181}$$

where $m_j^T = \dfrac{(V + \dot{d} - V_{Ej})^T}{|R_j|}\left[I - r_j\, r_j^T\right] = \dot{R}_j^T\,[I - r_j r_j^T]\,/|R_j|$

Now, note that equation (181) contains both $\dot{d}$ and $\delta\dot{d}$ terms as well as a δd term, (the $\dot{d}$ term is contained in m_j^T).

Looking first at $\dot{d}$, and remembering that $d = T_{C/A}\, d_A$, it is seen that

$$\dot{d} = \dot{T}_{C/A}\, d_A + T_{C/A}\frac{d}{dt}(d_A). \tag{182}$$

However, d_A is a constant vector in the aircraft reference frame, hence the time derivative of d_A is equal to zero. Also, $\dot{T}_{C/A} = [\,\omega_{A/C}{}^X]_C\, T_{C/A}$. Therefore equation (182) can be rewritten as

$$\dot{d} = [\,\omega_{A/C}X]_C \; T_{C/A} \; d_A = [\,\omega_{A/C}X]_C \; d \tag{183}$$

where $\omega_{A/C}$ = angular rate of the aircraft WRT the C frame, and

$[\omega_{A/C}X]_C$ = cross product matrix of $\omega_{A/C}$ expressed in C frame coordinates.

That portion of m_j^T which contains $\dot{d}$ can now be expressed as follows:

$$\frac{\dot{d}^T}{|R_j|}\left[I - r_j \, r_j^T\right] = \frac{[\omega_{A/C}X]^T}{|R_j|}\left[I - r_j \, r_j^T\right] \tag{184}$$

Now δd, as defined in Section 1, is given by:

$$\delta d = \delta T_{C/A} \; d_A$$

Taking the time derivative of this expression, and again noting that the time derivative of d_A in the aircraft frame is zero, yields

$$\delta\dot{d} = \delta\dot{T}_{C/A} \; d_A \tag{185}$$

Now $\delta T_{C/A}$ is a small angle transformation which is the sum a) of the ψ rotation and antenna/airframe flexure for a strapdown mechanization, and b) of ψ, plus flexure, plus platform attitude readout errors for stabilized platform mechanizations. Correspondingly $\delta\dot{T}_{C/A}$ is the sum of the time rate of change of each of these effects. If the update rate of the $T_{C/A}$ transformation is sufficiently fast to keep up with vehicle rotational motion, then the only $\delta\dot{d}$ error of consequence is that arising from antenna/airframe flexure rates. This flexure rate error is more properly modeled as time uncorrelated noise, considering the relatively long Kalman cycle time.

Assuming that the $\delta\dot{d}$ term is modeled as noise, (and now contained in the $n_{\dot{R}j}(t)$ term), then the M_j matrix can be defined using equation (181), and is given by:

$$M_j = \left[m_j^T \,\vdots\, r_j^T \,\vdots\, \text{ZERO} \,\vdots\, -m_j^T \,\vdots\, -r_j^T \,\vdots\, \text{ZERO} \,\vdots\, \text{UNITY} \,\vdots\, \text{ZERO} \right] \tag{186}$$

where the δd term has been neglected.

If the δd term is retained, then a (1 x 9) row vector, $m_j^T D$, is added to the M_j matrix and a (9 x 1) column vector $\delta T'_{A/C}$ is added to the error state vector. The matrix D, and the column vector $\delta T'_{A/C}$ are both defined in Section 1 of this appendix and m_j^T is defined in equation (181).

If operating in the C = EF frame, then [see equation (160)]:

$$|R_j| = |P + d - E_j| = |p + d - e_j| \tag{187}$$

and using equation (187), the range-rate is given by

$$\frac{d}{dt} |R_j| = \frac{d}{dt} |p + d - e_j| \tag{188}$$

From this equation, it can be seen that the same M_j matrix, defined by equation (186), will be arrived at where now the variables are computed with reference to the EF frame center and expressed in EF frame coordinates.

4. BAROMETRIC ALTIMETER MEASUREMENTS

The formulation of the M matrix for altitude measurements is based upon the geometry described in Figure 36.

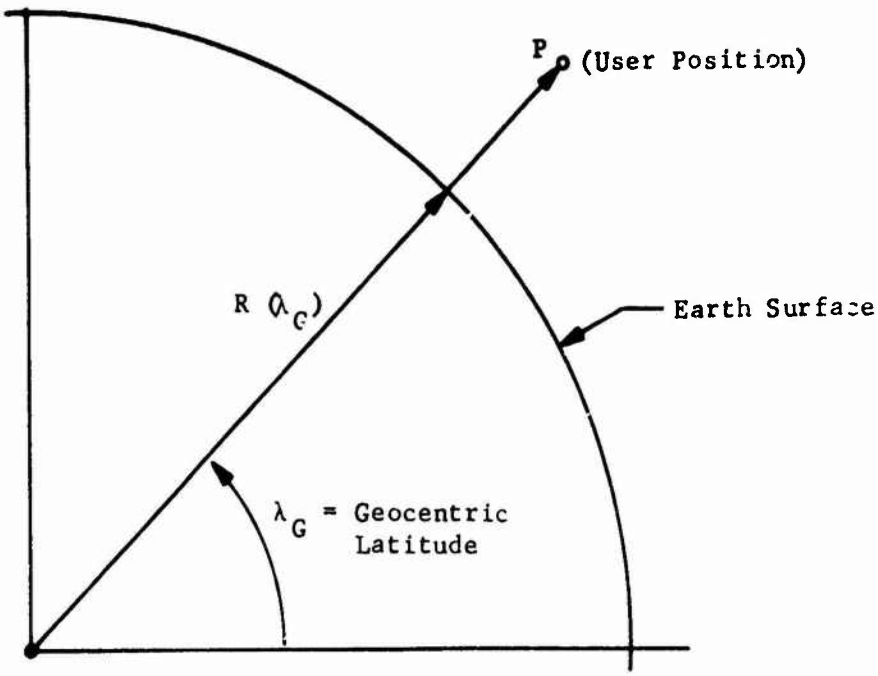


Figure 36. Altitude Measurement Geometry

Now let the vector h be defined as

$$h = P - R(\lambda_G) \tag{189}$$

This vector, h, differs from the actual vector altitude due to the eccentric anomaly. The difference in their magnitude is approximately one foot or less for altitudes up to 100,000 feet. Hence to a very good approximation, let the scalar altitude measurement be given as:

$$|h| = |P - R(\lambda_G)| = |P| - |R(\lambda_G)| \tag{190}$$

The terms $|P|$ and $|R(\lambda_G)|$ can be expressed as

$$|P(X,Y,Z)| = \sqrt{X^2 + Y^2 + Z^2} \tag{191}$$

and

$$|R(\lambda_G)| = \frac{|R_0|}{\left[1 + k \sin^2 \lambda_G\right]^{1/2}} \tag{192}$$

where $|R_0|$ = earth radius at the equator

$$k = \left[\frac{1}{(1 - e)^2} - 1\right]$$

$$e = \text{ellipticity} = 1/297.0$$

In the E frame, $\sin \lambda_G = Z/|P|$. Substituting this expression into equation (192) yields

$$|R\ (X,Y,Z)| = \frac{|R_0|}{\left[1 + k\left(\frac{Z^2}{X^2 + Y^2 + Z^2}\right)\right]^{1/2}} \tag{193}$$

To form the expression for $|\Delta h|$, let $|\hat{h}|$ be the computed value for $|h|$ and $|h_m|$ be the compensated measurement of $|h|$. Then subtract $|h_m|$ from $|\hat{h}|$ after expanding $|\hat{h}|$ in a Taylor series about the true value $|h|$. This results in the following expression:

$$|\Delta h| = \frac{\partial |P\ (X,Y,Z)|}{\partial P} \cdot \delta P - \frac{\partial |R\ (X,Y,Z)|}{\partial P} \cdot \delta P + \delta h_m + n_h(t) \tag{194}$$

where δh_m and $n_h(t)$ respectively denote the time correlated and time uncorrelated altitude reference measurement errors.

By definition,

$$\frac{\partial |P\ (X,Y,Z)|}{\partial P} \cdot \delta P = \frac{\partial |P|}{\partial X} \delta X + \frac{\partial |P|}{\partial Y} \delta Y + \frac{\partial |P|}{\partial Z} \delta Z$$

and

$$\frac{\partial |R\ (X,Y,Z)|}{\partial P} \cdot \delta P = \frac{\partial |R|}{\partial X} \delta X + \frac{\partial |R|}{\partial Y} \delta Y + \frac{\partial |R|}{\partial Z} \delta Z$$

Performing the indicated partial differentiations as indicated, and then gathering together like terms provides the following expressions:

$$\left(\frac{\partial|P|}{\partial X} - \frac{\partial|R|}{\partial X}\right) = \left(\frac{X}{|P|} - \frac{X}{|P|} k |R_0| \left|\frac{R}{R_0}\right|^3 \frac{Z^2}{|P|^3}\right)$$

$$\left(\frac{\partial|P|}{\partial Y} - \frac{\partial|R|}{\partial Y}\right) = \left(\frac{Y}{|P|} - \frac{Y}{|P|} k |R_0| \left|\frac{R}{R_0}\right|^3 \frac{Z^2}{|P|^3}\right)$$

$$\left(\frac{\partial|P|}{\partial Z} - \frac{\partial|R|}{\partial Z}\right) = \left(\frac{Z}{|P|} - \frac{Z}{|P|} k |R_0| \left|\frac{R}{R_0}\right|^3 \frac{Z^2 - |P|^2}{|P|^3}\right)$$

These results simplify based upon the following considerations:

$$\left(\frac{|R|}{|R_0|}\right) \leq 1 \qquad \left(\frac{|R|}{|R_0|}\right)^3 \leq 1$$

and $k = \dfrac{e\ (2 - e)}{(1-e)(1-e)} \cong 2e \cong 1/150$

Now, since $|P|$ and $|R_0|$ are of the same order of magnitude, X, Y, and Z are at most of the same order of magnitude as $|P|$, and since $2e \cong 1/150$, this implies that the second term in each of the above expressions is at least two orders of magnitude less than the first term. A very good approximation to equation (194) is arrived at by neglecting these terms; i.e.,

$$|\Delta h| = \left[\frac{\partial|P|}{\partial P} \,\vdots\, [\text{UNITY}]\right]\begin{bmatrix} \delta P \\ \text{----} \\ \delta h_m \end{bmatrix} + n_h(t) \tag{195}$$

where $\dfrac{\partial|P|}{\partial P} = \dfrac{\hat{X}}{|P|} \quad \dfrac{\hat{Y}}{|P|} \quad \dfrac{\hat{Z}}{|P|}$

For the case where the computational frame is the EF frame, some slight modifications result. The scalar $|h|$ is generated in the same fashion and again the partial of $|R|$ may be neglected. However, now

$$|P| = \sqrt{(x+x_c)^2 + (y+y_c)^2 + (z+z_c)^2} \tag{196}$$

where x_c, y_c, z_c are the components of the c vector (vector from origin of E frame to origin of EF frame) expressed in EF frame coordinates.

$$|\Delta h| = \left[\frac{\partial |P|}{\partial p} \;\vdots\; [\mathrm{UNITY}] \right] \begin{bmatrix} \delta p \\ --- \\ \delta h_m \end{bmatrix} + n_h(t) \tag{197}$$

where:

$$\frac{\partial |P|}{\partial p} = \quad \frac{\hat{x} + x_c}{|\hat{P}|} \quad \frac{\hat{y} + y_c}{|\hat{P}|} \quad \frac{\hat{z} + z_c}{|\hat{P}|}$$

APPENDIX VII

MEASUREMENT PREPROCESSING

This appendix summarizes the mathematics underlying established or promising approaches to four different types of measurement preprocessing module algorithms: measurement smoothing, measurement reasonableness testing, measurement selection, and measurement space averaging.

1. MEASUREMENT SMOOTHING

Denote the raw measurement at time t_i within the averaging interval by Y_i, and the smoothed measurement by $\overline{Y}$. Then:

$$\overline{Y} = (1/n) \sum_{i=1}^{n} Y_i \tag{198}$$

where n is the number of measurements Y_i obtained in the averaging interval. Next denote by M_i and V_i, respectively, the measurement matrix and the measurement noise corresponding to Y_i, such that:

$$Y_i = M_i X_i + V_i \tag{199}$$

where X_i is the system state at time t_i. Also:

$$X_i = \phi_{i,n} X_n - \int_{t_i}^{t_n} \phi (t_i, \tau) W(\tau) d\tau \tag{200}$$

where X_n is the state at time t_n (taken here for simplicity as corresponding to the end of the filter cycle), $\phi_{i,n}$ is the state transition matrix for the interval t_n to t_i, $\phi (t_i, \tau)$ is the state transition matrix for the interval τ to t_i, and $W(\tau)$ is random, forcing state noise.

Combining equations (198), (199) and (200) gives:

$$\overline{Y} = \overline{M}X_n + \overline{V} \tag{201}$$

where:

$$\overline{M} = (1/n)\sum_{i=1}^{n} M_i\ \phi_{i,n}$$

$$\overline{V} = (1/n)\sum_{i=1}^{n} V_i - (1/n)\sum_{i=1}^{n} \int_{t_i}^{t_n} M_i\ \phi\ (t_i,\ \tau)\ W(\tau)\ d\tau$$

The gain and the covariance matrix measurement update equations, which determine the new estimate and its associated covariance matrix from the prior estimate and the smoothed measurement Y, are derived using a procedure identical to that used in deriving the corresponding Kalman equations for simpler, end-of-interval measurements. Since this derivation is widely available in the literature, it is not repeated here. However, the results are:

$$b_K = (P\overline{M}^T - \overline{Z}^T)(\overline{M}P\overline{M}^T + \overline{C}' - \overline{M}\overline{Z} - \overline{Z}\ \overline{M}^T)^{-1} \tag{202}$$

$$P = P - b_K\ (\overline{M}P - \overline{Z}) \tag{203}$$

where b_K is the gain, P is the covariance matrix, and:

$$\overline{Z} = (1/n)\sum_{i=1}^{n} M_i'\ R_i \qquad M_i' = M_i\ \phi_{i,n} \tag{204}$$

$$R_i = \int_{t_i}^{t_n} \phi\ (t_n,\ \tau)\ K\phi^T\ (t_n,\ \tau)\ d\tau \tag{205}$$

$$\overline{C}' = (1/n^2)\sum_{i=1}^{n} \langle V_i\ V_i^T \rangle + (1/n^2)\sum_{i=1}^{n} \left[\sum_{j=1}^{n} (M_j'\ R_i\ M_i'^T + M_i'\ R_i\ M_j'^T) - M_i'\ R_i\ M_i'^T \right] \tag{206}$$

Note that for the special case of a single, end-of-the-interval measurement (i.e., $\overline{Y} = Y_1 = Y_n,\ t_i = t_n$), equations (202) and (203) reduce to the normal Kalman form ($M_i = M_n = \overline{M},\ \overline{Z} = 0$).

2. MEASUREMENT REASONABLENESS TESTING

Consider any single measurement Y. Denoting by $\hat{Y}$ the estimated value of that measurement based on the measurement matrix M and the existing state estimate $\hat{X}$, i.e.,

$$\hat{Y} = M\hat{X},$$

then the difference D, defined by

$$D = \hat{Y} - Y \tag{207}$$

is a convenient basis for a reasonableness test as follows. Using equation 4) it follows that

$$D = M\epsilon - V \tag{208}$$

where $\epsilon = \hat{X} - X$ is the error in the estimate $\hat{X}$.

Taking the expected value of the square of D therefore gives

$$< D^2> \;=\; < DD^T> \;= MPM^T + C \tag{209}$$

An attractive reasonableness test can now be formulated as

$$D^2 \leq k^2 \; (MPM^T + C) \rightarrow Y \text{ is reasonable}$$

$$D^2 > k^2 \; (MPM^T + C) \rightarrow Y \text{ is not reasonable} \tag{210}$$

where k is the number of standard deviations selected as the reasonableness limit for $|D|$.

3. MEASUREMENT SELECTION

Denote by δ and K, respectively, a generalized miss vector and its (sensitivity coefficient) matrix relationship to ϵ, the error in the state estimate $\hat{X}$; i.e.,

$$\delta = K\epsilon \tag{211}$$

The expected squared radial miss distance is just the trace of the miss covariance matrix:

$$KPK^T \tag{212}$$

Further, denoting by P and P_{Aj}, respectively, the estimate error covariance matrix just before and just after use of a particular measurement Y_j (measurement matrix M_j and measurement noise C_j), then the change in the expected squared radial miss distance is just:

$$K(P_{Aj} - P)\, K^T \tag{213}$$

But since (assuming for simplicity that all measurements Y_j are scalar):

$$P_{Aj} - P = -\frac{PM_j^T\, M_j P}{M_j\, PM_j^T + C_j} \tag{214}$$

it follows that the trace of the matrix Δ_j, defined by:

$$\Delta_j = \frac{KPM_j^T\, M_j\, PK^T}{M_j\, PM_j^T + C_j} \tag{215}$$

is the decrease in the expected squared radial miss distance.

An attractive measurement selection algorithm might therefore be based simply on calculating the trace of Δ_j for each of the available measurements, and selecting for further processing (i.e., processing by the Estimation and Control Module) that measurement which yielded the largest trace value.

4. MEASUREMENT SPACE AVERAGING

Consider a set of available (scalar) measurement differences Y_j ($j = 1,2,\ldots,n$). Denote the corresponding set of unit line-of-sight vectors by r_j ($j = 1,2,\ldots,$ n)*. Then the space-averaged measurement on this set is defined by:

$$\boxed{\overline{Y} = 1/n \sum_{j=1}^{n} r_j Y_j} \tag{216}$$

---

* In the case of earth-mode emitters, replace r_j by τ_j (see the symbol glossary).

Define the error in Y_j by δY_j, and that portion of the measurement matrix associated with the modeled overall system errors by $\hat{M}_j$. Also let x denote the modeled error state vector. Then:

$$Y_j = \hat{M}_j x + \delta Y_j \tag{217}$$

where δY_j consists in all the unmodeled measurement-difference errors, plus noise. Using equation (217) in equation (216) leads to:

$$\overline{Y} = \overline{M}x + \delta Y \tag{218}$$

where:

$$\boxed{\overline{M} = 1/n \sum_{j=1}^{n} r_j M_j} \tag{219}$$

and

$$Y = 1/n \sum_{j=1}^{n} \delta Y_j \tag{220}$$

Remembering that the errors δY_j are assumed to be unmodeled, then if the scalar measurements Y_j were processed through the filter sequentially one at a time, the filter would essentially correct user position along each line of sight in turn, in each instance with an error essentially equal to $r_j \, \delta Y_j$. These errors would be cumulative, so that the final user position error δP_{SQ} after sequential, one-at-a-time processing of all n measurements Y_j would be given by the expression:

$$\delta P_{SQ} = \sum_{j=1}^{n} r_j \, \delta Y_j \tag{221}$$

On the other hand, use of the space-averaged measurement defined by equation (216) would essentially result in only the user position error:

$$\delta P_{SA} = \delta Y = 1/n \, \delta P_{SQ} \tag{222}$$

Thus, δP_{SA} will in general be much smaller than δP_{SQ}; more specifically, the expected value of δP_{SA} is in fact n times smaller than that of δP_{SQ}.

APPENDIX VIII

RECURSION EQUATIONS FOR MEASUREMENT TIME SMOOTHING AND PREDICTION MATRICES

The Phase I functional formulation of the Kalman Filter Module provided closed-form equations for generating the sets of matrices associated with both the measurement averaging and the estimate and covariance matrix prediction operations.* However, since in many practical cases it may be necessary to update these matrices at much higher rates than the overall Kalman Filter Module execution rate, recursive rather than closed-form formulations are preferable. This appendix presents and discusses an appropriate set of such recursive formations.**

The matrices (and their closed-form expressions) under consideration here are:

$$\bar{Y}_n = \frac{1}{n} \sum_{i=1}^{n} Y_i \tag{223}$$

$$\bar{C}_n = \frac{1}{n} \sum_{i=1}^{n} C_i \tag{224}$$

$$\bar{M}_{n,F} = \frac{1}{n} \sum_{i=1}^{n} M_i \Phi_{i,F} \tag{225}$$

$$\bar{N}_{n,F} = \frac{1}{n} \sum_{i=1}^{n} M_i \Phi_{i,F} G_{F,i} \tag{226}$$

*See Appendix VII of this (Phase II) report.

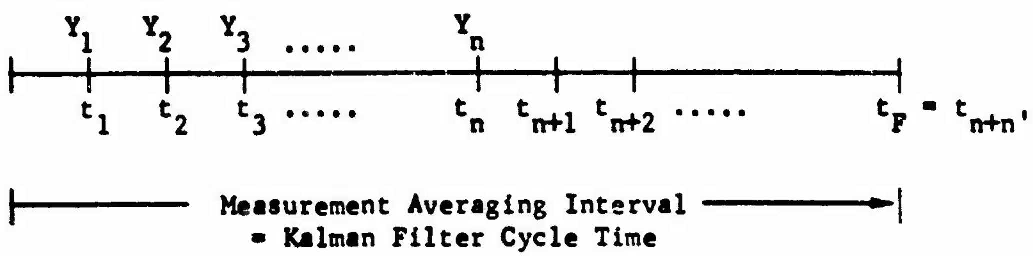
**The derivations, which in many cases are rather lengthy, are omitted here for brevity and clarity. The recursive formulae presented here, however, can be easily verified by direct substitution of the closed-form formulae.

$$\bar{Z}_{n,F} = \frac{1}{n}\sum_{i=1}^{n} M_i \phi_{i,F} R_{F,i} \tag{227}$$

$$\bar{W}_{n,F} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} M_{i,F} R_{F,ij*} M^T_{j,F} \tag{228}$$

$(M_{i,F} = M_i \phi_{i,F}, \quad ij* = \text{larger of } i \text{ and } j)$

Note that formulae (225) through (228) represent an important generalization of the corresponding formulae presented in the Phase I final report, which are valid only for generating a new Kalman estimate of the system error state for the time at which the last measurement Y_n was taken. On the other hand, the generalized formulae given here apply in the more general case when a delay exists between the time of the last measurement and the time t_F associated with the Kalman estimate (see the time sequence sketch below).



The recursive formulations of the closed-form expressions above are:

$$\bar{Y}_i = \bar{Y}_{i-1} + \frac{1}{i}\left(Y_i - \bar{Y}_{i-1}\right) \quad \left(\bar{Y}_o = 0,\ i = 1 \text{ to } n\right) \tag{229}$$

$$\bar{C}_i = \bar{C}_{i-1} + \frac{1}{i}\left(C_i - \bar{C}_{i-1}\right) \quad \left(\bar{C}_o = 0,\ i = 1 \text{ to } n\right) \tag{230}$$

$$\left\{\begin{array}{l} \overline{M}_i = \overline{M}_{i-1}\phi_{i-1,i} + \frac{1}{i}\left(M_i - \overline{M}_{i-1}\phi_{i-1,i}\right) \quad \left(\overline{M}_o = 0,\ i = 1 \text{ to } n\right) \\ \phi^{*}_{n,n+i'} = \phi_{n,n+i'-1}\phi_{n+i'-1,n+i'}\left(\phi_{n,n} = I,\ i' = 1 \text{ to } n'\right) \\ \overline{M}_{n,F} = \overline{M}_n\phi_{n,F} \end{array}\right\} \qquad (231)$$

$$\left\{\begin{array}{l} \overline{N}_i = \left(1 - \frac{1}{i}\right)\left(\overline{N}_{i-1} + \overline{M}_{i-1}G_{i-1,i}\right) \quad \left(\overline{N}_o = 0,\ i = 1 \text{ to } n\right) \\ G^{**}_{n,n+i'} = G_{n,n+i'-1} + \phi_{n,n+i'-1}G_{n+i'-1,n+i'}\left(G_{n,n} = 0,\ i' = 1 \text{ to } n'\right) \\ \overline{N}_{n,F} = \overline{N}_n - \overline{M}_nG_{n,F} \end{array}\right\} \qquad (232)$$

$$\left\{\begin{array}{l} \overline{Z}_i = \left(1 - \frac{1}{i}\right)\left(\overline{Z}_{i-1}\phi^T_{i,i-1} + \overline{M}_{i-1}R_{i,i-1}\right) \quad \left(\overline{Z}_o = 0,\ i = 1 \text{ to } n\right) \\ R^{***}_{n+i',n} = \phi_{n+i',n+i'-1}R_{n+i'-1,n}\phi^T_{n+i',n+i'-1} \\ \qquad\qquad + R_{n+i',n+i'-1}\left(R_{n,n} = 0,\ i' = 1 \text{ to } n'\right) \\ \overline{Z}_{n,F} = \overline{Z}_n\phi^T_{F,n} + \overline{M}_{n,F}R_{F,n} \end{array}\right\} \qquad (233)$$

$$\left\{\begin{array}{l} \overline{W}_i = \left(1 - \frac{1}{i}\right)^2\overline{W}_{i-1} + \left(\overline{M}_i - \frac{1}{i}M_i\right)R_{i,i-1}\left(\overline{M}_i - \frac{1}{i}M_i\right)^T \\ \qquad \left(\overline{W}_o = 0,\ i = 1 \text{ to } n\right) \\ \overline{W}_{n,F} = \overline{W}_n + \overline{M}_{n,F}R_{F,n}\overline{M}^T_{n,F} \end{array}\right\} \qquad (234)$$

Formulae (229) through (234) comprise the recursive formulation for the measurement averaging matrices defined by equations (223) through (228). The corresponding recursive formulae for prediction are:

---

*This formula is for recursive generation of $\phi_{n,F}$ for use in the $\overline{M}_{n,F}$ formula.

**This formula is for recursive generation of $G_{n,F}$ for use in the $\overline{N}_{n,F}$ formula.

***This formula is for recursive generation of $R_{F,n}$ for use in the $\overline{Z}_{n,F}$ and and $\overline{W}_{n,F}$ formulae.

$$\left\{\begin{aligned} \hat{X}_i &= \phi_{i,i-1}\hat{X}_{i-1} + G_{i,i-1}\dot{u} \\ P_i &= \phi_{i,i-1}P_{i-1}\phi^T_{i,i-1} + R_{i,i-1} \end{aligned}\right\} \tag{235}$$

Use of the above recursive formulae (229) through (235) requires definition of the fundamental incremental matrices $\phi_{i-1,i}$ (and $\phi_{n+i'-1,n+i'}$), $G_{i-1,i}$ (and $G_{n+i'-1,n+i'}$), and $R_{i,i-1}$ (and $R_{n+i',n+i'-1}$). It can be demonstrated that*

$$\phi_{i-1,i} = I + \int_{t_i}^{t_{i-1}} A(u)\ \phi\left(u,t_i\right)du$$

$$G_{i-1,i} = \left(t_{i-1} - t_i\right)I + \int_{t_i}^{t_{i-1}} A(u)G\left(u,t_i\right)du$$

$$R_{i,i-1} = \left(t_i - t_{i-1}\right)K + \int_{t_{i-1}}^{t_i}\left\{A(u)R\left(u,t_{i-1}\right) + R^T\left(u,t_{i-1}\right)A^T(u)\right\}du$$

If the time interval t_i-t_{i-1} is small, then these formulae can be approximated by:

$$\left\{\begin{aligned} \phi_{i-1,i} &\approx I - A_{i-1}\left(t_i - t_{i-1}\right) \\ G_{i-1,i} &\approx -I\left(t_i - t_{i-1}\right) \\ R_{i,i-1} &\approx K\left(t_i - t_{i-1}\right) \\ \phi_{i,i-1} &= \phi^{-1}_{i-1,i} \approx I + A_{i-1}\left(t_i - t_{i-1}\right) \\ G_{i,i-1} &= -\phi_{i,i-1}G_{i-1,i} \approx I\left(t_i - t_{i-1}\right) \\ R_{i-1,i} &= -\phi_{i,i-1}R_{i,i-1}\phi^T_{i,i-1} \approx -K\left(t_i - t_{i-1}\right) \end{aligned} \quad \begin{pmatrix} i = 1 \text{ to } n \\ \text{and} \\ i = 1 \text{ to } n' \end{pmatrix}\right\} \tag{236}$$

*Again, the derivations are omitted here for brevity, but the formulae can easily be verified by using the defining integral and differential equations for ϕ, G, and R.

APPENDIX IX

PROCESSOR DESIGN FOR
NONLINEAR LOS PSEUDORANGING SITUATIONS

An important multilateration processor design problem area centers on the proper use of available radio navigation data in those prospective operational situations where the user/emitter relative position uncertainties are comparable in size to the actual user/emitter ranges themselves. The two principal such situations considered here--both against the background of the assumed availability of a net of LOS radio links only--are (a) navigation start-up (i.e., user position and velocity initialization) when no (or only coarse) a priori position and velocity estimates are available, and (b) transition from globally referenced (E frame) to locally referenced (EF frame) navigation on acquisition of an objective area LOS net. In the former case a large user position error, and in the latter case both a large user position error (e.g., from say pure inertial enroute navigation) and a large emitter net position error (e.g., from say an objective area datum plane uncertainty) are present which may well be comparable in size to the user/emitter ranges themselves.

This appendix first formulates and discusses the mathematical basis of the problem, and then outlines some promising candidate algorithms for its solution.

1. GENERAL DISCUSSION

To attack the problem, consider first the fundamental ranging equation, which relates the primary navigational entities involved in pseudoranging -- which includes two-way ranging as a subcase -- between the user and the jth emitter (see Figure 37).

$$|R_j| = |P - E_j|* \qquad (237)$$

*Throughout these notes, the symbols P, δP, E_j and δE can everywhere be replaced with p, δp, e_j, and δe_j, respectively; i.e., all results are equally valid for either E or EF frame referenced computations.

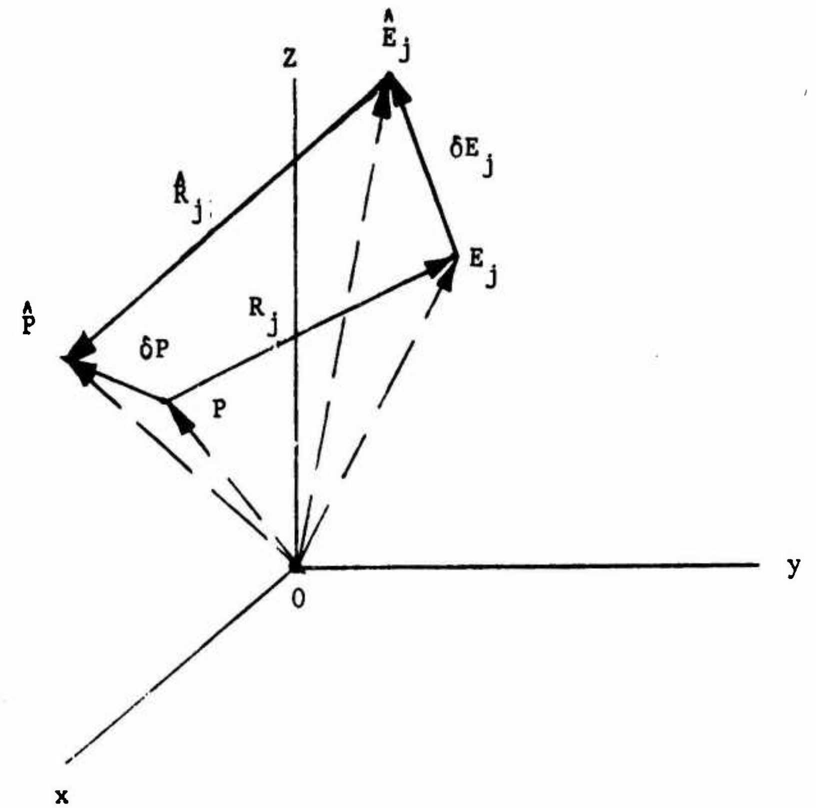


Figure 37. User-Emitter Geometry

where P, E_j, and R_j are respectively the true user position vector, the true jth emitter position vector, and the true jth-emitter-to-user range vector. The left side of equation (237) can be written:

$$|R_j| = R_{mj} - \delta R_{mb} - \delta R_{mNj} \tag{238}$$

where R_{mj} = radio pseudorange (phase of user-received emitter signal relative to user clock phase reference)

δR_{mb} = difference between user and emitter clock phase references

δR_{mNj} = noise in R_{mj}

Correspondingly, the right side of equation (237) can be written:

$$|P-E_j| = |\hat{R}_j - \delta R_j| \tag{239}$$

where $\hat{R}_j = \hat{P} - \hat{E}_j$, $\delta R_j = \delta P - \delta E_j$, and the superior hat implies the estimated, rather than the true, value of the hatted quantity. Equating these results according to equation (237) gives:

$$R_{mj} - \delta R_{mb} - \delta R_{mNj} = |\hat{R}_j - \delta R_j| \tag{240}$$

Squaring both sides of equation (240), using equation (238), and collecting terms leads to the fundamental, <u>nonlinear</u>, pseudoranging measurement/state relationship:

$$Y_{Rj} = M_{Rj} x + v_{Rj} + Q_{Rj} \tag{241}$$

where:

$$x^T = [\, \delta P^T \,\vdots\, \delta \dot{P}^T \,\vdots\, \delta R_{mb} \,\vdots\, \delta \dot{R}_{mb} \,\vdots\, \ldots \,\vdots\, \delta E_j^T \,\vdots\, \ldots \,\vdots\, \delta \dot{E}_j^T \,\vdots\, \ldots]$$

$$M_{Rj} = [\hat{R}_j^T \,\vdots\, 0 \,\vdots\, -R_{mj} \,\vdots\, 0 \,\vdots\, \ldots \,\vdots\, -\hat{R}_j^T \,\vdots\, \ldots \,\vdots\, 0 \,\vdots\, \ldots]$$

$$v_{Rj} = -\left(|R_j| + \frac{1}{2}\delta R_{mNj}\right)\delta R_{mNj}$$

$$Y_{Rj} = \frac{1}{2}\left(|\hat{R}_j|^2 - R_{mj}^2\right) \qquad (\hat{R}_j = \hat{P}_j - \hat{E}_j)$$

$$Q_{Rj} = \frac{1}{2}\left(\delta R_{mb}^2 - \delta R_j^T \delta R_j\right) \qquad (\delta\dot{P} = \delta v)$$

A second fundamental, nonlinear, pseudorange-rating equation is obtained by time differentiating (240), multiplying the resulting equation by (240) itself, and collecting terms. The result is.

$$Y_{RRj} = M_{RRj} x + v_{RRj} + Q_{RRj} \tag{242}$$

where:

$$x^T = [\delta P^T \vdots\ \delta \dot{P}^T \vdots\ \delta R_{mb} \vdots\ \delta \dot{R}_{mb} \vdots\ \ldots \vdots\ \delta E_j^T \vdots\ \ldots \vdots\ \delta \dot{E}_j^T \vdots\ \ldots]$$

$$M_{RRj} = [\dot{\hat{R}}_j^T \vdots\ \hat{R}_j^T \vdots\ -\dot{R}_{mj} \vdots\ -R_{mj} \vdots\ \ldots \vdots\ -\dot{\hat{R}}_j^T \vdots\ \ldots \vdots\ -\hat{R}_j^T \vdots\ \ldots]$$

$$V_{RRj} = \dot{V}_{Rj}$$

$$Y_{RRj} = \dot{Y}_{Rj} = \hat{R}_j^T \dot{\hat{R}}_j - R_{mj}\,\dot{R}_{mj} \qquad (\dot{\hat{R}}_j = \dot{\hat{P}}_j - \dot{\hat{E}}_j)$$

$$Q_{RRj} = \dot{Q}_{Rj} = \delta R_{mb}\,\delta \dot{R}_{mb} - \delta R_j^T\,\delta \dot{R}_j$$

and

$\dot{R}_{mj}$ = radio pseudo range rate (phase rate of user-received emitter signal relative to user clock phase reference)

$\delta \dot{R}_{mb}$ = time rate of change of δR_{mb}

Equations (241) and (242) are the fundamental nonlinear relations between the measurements Y_{Rj} (or Y_{RRj}) and the error state x; the nonlinearities are in particular grouped in the term Q_{Rj} (or Q_{RRj}). These equations are the basis for the development of several, completely linear measurement/ state relationships in what follows.

Before proceeding, it should first be noted that if the conditions

$$|\delta R_{mb}| << |R_{mj}|$$
$$|\delta R_j| << |\hat{R}_j| \qquad (243)$$

hold, then the nonlinearities in equations (241) and (242) are negligible, and these equations then reduce to the linear relationships:

$Y_{Rj} = M_{Rj}x + v_{Rj}$	Negligible Nonlinearities	(244)
$Y_{RRj} = M_{RRj}x + v_{RRj}$	One emitter; pseudo ranging or pseudo ranging and pseudo range rating.	(245)

Equations (244) and (245) are equivalent to the linear measurement/state relationships developed in the first phase of the multilateration processor development effort, and which are derived and presented in Appendix VI of this Phase II final report.

Those equations, like (8) and (9) above, are also valid only when the conditions (7) hold.

Returning now to the consideration of the basic nonlinear equations (241) and (242), consider first the case when two or more emitters are simultaneously available. It is evident from the form of equation (241) and of Q_{Rj} that if:

$$\delta E_j = \delta E, \text{ or } \delta E_j = 0 \tag{246}$$

then writing down equation (241) for emitters j and k and differencing the resulting equations leads to the linear form:

$$\Delta Y_{Rjk} = \Delta M_{Rjk}x + \Delta v_{Rjk} \tag{247}$$

Pseudo Ranging; two or more emitters, condition (10) holds

where:

$$\Delta Y_{Rjk} = Y_{Rj} - Y_{Rk}$$

$$\Delta M_{Rjk} = M_{Rj} - M_{Rk}$$

Significant Nonlinearities

$$\Delta v_{Rjk} = v_{Rj} - v_{Rk}$$

where it is understood that the set of variables δE_j in x has been collapsed to either the single common (datum plane) error vector δE, or is obviated entirely if $\delta E_j = 0$.

Similarly, if (246) holds, the same procedure with equation (242) also leads to:

$\Delta Y_{RRjk} = \Delta M_{RRjk} x + \Delta v_{RRjk}$	<u>Pseudo ranging and pseudo range rating;</u>	(248)
where:		
$\Delta Y_{RRjk} = Y_{RRj} - Y_{RRk}$	<u>Two or more emitters; condition (246) holds</u>	
$\Delta M_{RRjk} = M_{RRj} - M_{RRk}$	<u>Significant Nonlinearities</u>	
$\Delta v_{RRjk} = v_{RRj} - v_{RRk}$		

Consider next the case where only <u>one</u> emitter is available. Under these conditions, dropping the unnecessary emitter subscript j, and introducing instead the time subscript i, equations (5) and (5) can be written:

$$Y_{Ri} = M_{Ri} x_i + v_{Ri} + Q_{Ri} \tag{249}$$

$$Y_{RRi} = M_{RRi} x + v_{RRi} + Q_{RRi} \tag{250}$$

Assume now that:

$$\begin{aligned} \delta\dot{P} &= \text{constant } (\delta\dot{R} = \text{constant}) \\ \delta\dot{E} &= \text{constant} \\ \delta\dot{R}_{mb} &= \text{constant} \end{aligned} \tag{251}$$

Therefore:

$$Q_{Ri} = \frac{1}{2}\left(\delta R_{mbi}^2 - \delta R_i^T\,\delta R_i\right)$$

$$Q_{RRi} = \delta\dot{R}_{mb}\,\delta R_{mbi} - \delta\dot{R}^T\,\delta R_i$$

It is obvious by inspection that simple differencing of equation (249) [or equation (250)] at two different time points will not remove these nonlinear terms. However, given the assumptions (251), higher-order differencing will. To this end, assume that some linear combination L_R of the measurements Y_{Ri}, and some (other) linear combination L_{RR} of the measurements Y_{RRi} can be identified such that:

$$L_R Q_{Ri} = \sum_{i=1}^{n_R} w_{Ri} Q_{Ri} = 0 \tag{252}$$

$$L_{RR} Q_{RRi} = \sum_{i=1}^{n_{RR}} w_{RRi} Q_{RRi} = 0 \tag{253}$$

where w_{Ri} and w_{RRi} are the weights associated with the linear combinations and n_R and n_{RR} are the required number of measurements which must be combined. To find these, note that:

$$\begin{aligned} Q_{Ri} &= \frac{1}{2}\left[\left(\delta R_{mbn} + \delta\dot{R}_{mb}\,\Delta t_{in}\right)^2 - \left|\delta R_n + \delta\dot{R}\,\Delta t_{in}\right|^2\right] \\ &= \frac{1}{2}\left[\left(\delta R^2_{mbn} + 2\delta\dot{R}_{mb}\,\delta R_{mbn}\,\Delta t_{in} + \delta\dot{R}^2_{mb}\,\Delta t^2_{in}\right)\right. \\ &\quad \left. - \left|\delta R_n\right|^2 - 2\delta R_n^T \delta\dot{R}\Delta t_{in} - \left|\delta\dot{R}\right|^2 \Delta t^2_{in}\right] \end{aligned} \tag{254}$$

$$\begin{aligned} \text{or } Q_{Ri} &= \frac{1}{2}\left(\delta R^2_{mbn} - \left|\delta R_n\right|^2\right) + \left(\delta\dot{R}_{mb}\,\delta R_{mbn} - \delta R_n^T \delta\dot{R}\right)\Delta t_{in} \\ &\quad + \frac{1}{2}\left(\delta\dot{R}_{mb}^{\,2} - \left|\delta\dot{R}\right|^2\right)\Delta t^2_{in} \end{aligned} \tag{255}$$

and

$$Q_{RRi} = \delta\dot{R}_{mb}\left(\delta\dot{R}_{mbn} + \delta\dot{R}_{mb}\Delta t_{in}\right) - \delta\dot{R}^T\left(\delta R_n + \delta\dot{R}\Delta t_{in}\right)$$

$$\text{or } Q_{RRi} = \left(\delta\dot{R}_{mb}\delta R_{mbn} - \delta\dot{R}^T\,\delta R_n\right) + \left(\delta\dot{R}^2_{mb} - \left|\delta\dot{R}\right|^2\right)\Delta t_{in} \tag{256}$$

where $\Delta t_{in} = t_i - t_n$.

$L_R\,Q_{Ri}$ and $L_{RR}\,Q_{RRi}$ therefore have the form:

$$L_R Q_{Ri} = A_{R1}\sum_{i=1}^{n_R} w_{Ri} + A_{R2}\sum_{i=1}^{n_R} w_{Ri}\Delta t_{in} + A_{R3}\sum_{i=1}^{n_R} w_{Ri}\Delta t_{in}^2 \tag{257}$$

$$L_{RR} Q_{RRi} = A_{RR1}\sum_{i=1}^{n_{RR}} w_{RRi} + A_{RR2}\sum_{i=1}^{n_{RR}} w_{RRi}\Delta t_{in} \tag{258}$$

where the A_R's and A_{RR}'s are independent of the summation index i. Since these are arbitrary coefficients, $L_R\,Q_{Ri}$ and $L_{RR}\,Q_{RRi}$ can be zero only if:

$$\sum_{i=1}^{n_R} w_{Ri} = \sum_{i=1}^{n_R} w_{Ri}\Delta t_{in} = \sum_{i=1}^{n_R} w_{Ri}\Delta t_{in}^2 = 0 \tag{259}$$

$$\sum_{i=1}^{n_{RR}} w_{RRi} = \sum_{i=1}^{n_{RR}} w_{RRi}\Delta t_{in} = 0 \tag{260}$$

To simplify these conditions, assume that the measurements are equi-time-spaced, Δt apart. Then, since $\Delta t_{in} = (i-n)\,\Delta t$, these conditions reduce to:

$$\sum_{i=1}^{n_R} w_{Ri} = \sum_{i=1}^{n_R} i w_{Ri} = \sum_{i=1}^{n_R} i(2n-i)\, w_{Ri} = 0 \tag{261}$$

$$\sum_{i=1}^{n_{RR}} w_{RRi} = \sum_{i=1}^{n_{RR}} i w_{RRi} = 0 \tag{262}$$

The smallest values of n_R and n_{RR} for which these can be satisfied are $n_R = 3$ and $n_{RR} = 2$. Based on these values, the resulting weights are:

$w_{R1} = k_R$	$w_{RR1} = k_{RR}$
$w_{R2} = 3k_R$	$w_{RR2} = -2k_{RR}$
$w_{R3} = 3k_R$	$w_{RR3} = k_{RR}$
$w_{R4} = -k_R$	

where k_R and k_{RR} are arbitrary constants. For simplicity, take $k_R = k_{RR} = 1$.

Next note that $x_i = \Phi_{i,n} x_n$, where, because of the conditions (251), the transition matrix $\Phi_{i,n}$* is given by:

$$\Phi_{i,n} = \left[\begin{array}{c:c:c:c:c:c} I & (i-n)\Delta t I & 0 & 0 & 0 & 0 \\ \hdashline 0 & I & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 1 & (i-n)\Delta t & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & I & (i-n)\Delta t I \\ \hdashline 0 & 0 & 0 & 0 & 0 & I \end{array}\right] \tag{263}$$

where

$$X^T = \left[\delta P^T \,\vdots\, \delta \dot{P}^T \,\vdots\, \delta R_{mb} \,\vdots\, \delta \dot{R}_{mb} \,\vdots\, \delta E^T \,\vdots\, \delta \dot{E}^T\right]$$

Therefore, equations (249) and (250) can be written:**

$$Y_{Ri} = M_{Ri}\, \Phi_{i,n_R} x_{n_R} + v_{Ri} + Q_{Ri} \tag{264}$$

$$Y_{RRi} = M_{RRi}\, \Phi_{i,n_{RR}} x_{n_{RR}} + v_{RRi} + Q_{RRi} \tag{265}$$

Operating on these with L_R and L_{RR} respectively and remembering that $L_R\, Q_{Ri} = L_{RR}\, Q_{RRi} = 0$, one can write the desired linear measurement/state forms:

$$\sum_{i=1}^{4} w_{Ri} Y_{Ri} = \left(\sum_{i=1}^{4} w_{Ri} M_{Ri} \Phi_{i,4}\right) x_4 + \sum_{i=1}^{4} w_{Ri} v_{Ri} \tag{266}$$

$$\sum_{i=1}^{4} w_{RRi} Y_{RRi} = \left(\sum_{i=1}^{3} w_{RRi} M_{RRi} \Phi_{i,3}\right) x_3 + \sum_{i=1}^{3} w_{RRi} v_{RRi} \tag{267}$$

* I.e., for transition of the state x from time t_n to time t_i.

**This result tacitly assumes that no forcing state noise of consequence acts on x in the time period of interest here.

These can be finally summarized in vector/matrix form as:

$$L_{R4}Y_{Ri} = \mu_{R4}x_4 + L_{R4}v_{Ri} \qquad (268)$$

$$L_{RR3}Y_{RRi} = \mu_{RR3}x_3 + L_{RR3}v_{RRi} \qquad (269)$$

<u>One emitter; significant nonlinearities; pseudo ranging or pseudo ranging and pseudo range rating</u>

where

$$L_{R4}Y_{Ri} = w^T_{R4S}Y_{R4S} \qquad Y^T_{R4S} = \begin{bmatrix} Y_{R1} & Y_{R2} & Y_{R3} & Y_{R4} \end{bmatrix}$$

$$L_{RR3}Y_{RRi} = w^T_{RR3S}Y_{RR3S} \qquad Y^T_{RR3S} = \begin{bmatrix} Y_{RR1} & Y_{RR2} & Y_{RR3} \end{bmatrix}$$

$$L_{R4}v_{Ri} = w^T_{R4S}v_{R4S} \qquad v^T_{R4S} = \begin{bmatrix} v_{R1} & v_{R2} & v_{R3} & v_{R4} \end{bmatrix}$$

$$L_{RR3}v_{RRi} = w^T_{RR3S}v_{RR3S} \qquad v^T_{RR3S} = \begin{bmatrix} v_{RR1} & v_{RR2} & v_{RR3} \end{bmatrix}$$

$$\mu_{R4} = \begin{bmatrix} w^T_{R4S}\hat{R}^T_{4S} & -3\Delta t w^T_{RR3S}\hat{R}^T_{3S} & -w^T_{R4S}R_{m4S} & 3\Delta t w^T_{RR3S}R_{m3S} & -w^T_{R4S}\hat{R}^T_{4S} & 3\Delta t w^T_{RR3S}\hat{R}^T_{3S} \end{bmatrix}$$

$$\mu_{R3} = \begin{bmatrix} w^T_{RR3S}\dot{\hat{R}}^T_{3S} & w^T_{RR3S}\hat{R}^T_{3S} & -w^T_{RR3S}\dot{R}_{m3S} & -w^T_{RR3S}R_{m3S} & -w^T_{RR3S}\dot{\hat{R}}^T_{3} & -w^T_{RR3S}\hat{R}^T_{3S} \\ & -2w^T_{RR2S}\dot{\hat{R}}^T_{2S} & & +2w^T_{RR2S}\dot{R}_{m2S} & & +2w^T_{RR2S}\dot{\hat{R}}^T_{2S} \end{bmatrix}$$

and

$$w^T_{R4S} = [1\ -3\ \ 3\ \ 1\ ],\ w^T_{RR3S} = [1\ -\ 2\ \ 1],\ w^T_{RR2S} = [1\ -\ 1]$$

$$\hat{R}_{4S} = [\hat{R}_1 \vdots \hat{R}_2 \vdots \hat{R}_3 \vdots \hat{R}_4],\ \hat{R}_{3S} = [\hat{R}_1 \vdots \hat{R}_2 \vdots \hat{R}_3],\ \hat{R}_{2S} = [\hat{R}_1 \vdots \hat{R}_2]$$

$$R^T_{m4S} = [R_{m1}\ \ R_{m2}\ \ R_{m3}\ \ R_{m4}],\ R^T_{m3S} = [R_{m1}\ R_{m2}\ R_{m3}]$$

$$\dot{\hat{R}}_{3S} = [\dot{\hat{R}}_1 \vdots \dot{\hat{R}}_2 \vdots \dot{\hat{R}}_3],\ \dot{\hat{R}}_{2S} = [\dot{\hat{R}}_1 \vdots \dot{\hat{R}}_2]$$

$$\dot{R}^T_{m3S} = [\dot{R}_{m1S} \vdots \dot{R}_{m2S} \vdots \dot{R}_{m3S}],\ \dot{R}^T_{m2S} = [\dot{R}_{m1S} \vdots \dot{R}_{m2S}]$$

2. CANDIDATE ALGORITHMS

Depending on whether only one, or two or more LOS emitter links are simultaneously available, equations (268), (269), or (247), (248) respectively furnish the basis for any of a wide variety of types of linear estimators of the error state x.

For example, the estimator selected could be of the closed form, batch type on the one hand, or of the recursive (iterative) form on the other. Further, it might either be of the least-squares type, or alternatively of the statistical (e.g., minimum variance, maximum likelihood, etc.) type.

However, the fact that (recursive, minimum variance) Kalman filtering has already been selected in Phase I as the best technique for processing the standard, linearized, one-emitter-at-a-time, LOS measurements [as defined in AFAL-TR-72-80, or equivalently by equations (244) and (245)], militates for the selection of this same technique for processing the special-case, start-up and reference frame transition measurements derived in equations (268), (269) or (247), (248).

Such a selection ensures maximum compactness and efficiency of the overall processor program, because of the large body of Kalman Filter Module subroutines which can then be developed and used in common for processing the standard type, and the special-case types (hereafter called the start-up type) of measurement alike. In fact, the only important difference associated with processing these two different types of measurement lies in the Kalman Filter Measurement Preprocessing Submodule, where in start-up situations, measurements and measurement matrices which are essentially just simple linear combinations of the

standard measurements and measurement matrices must be constructed for use by the Estimation and Control Submodule.*

With this approach, the appropriate times of transition between use of the start-up and the standard types of measurement might be specifiable by means of a simple test based on the variance levels of δR and δR_{mb}, using appropriate elements from the Kalman Filter covariance matrix, as follows.

A suitable starting point for defining such a test has already been identified by the conditions (243), or equivalently:

$$\left|\delta R_{mb}\right| \le k \left|R_{mj}\right| \qquad (0 < k << 1) \tag{270a}$$

$$\left|\delta R\right| \le k \left|\hat{R}_j\right| ** \tag{270b}$$

Squaring gives:

$$\delta R_{mb}^2 \le k^2 R_{mj}^2 \tag{271a}$$

$$\delta R^T \delta R \le k^2 \hat{R}_j^T \hat{R}_j \tag{271b}$$

Taking expected values of these gives:***

$$\langle \delta R_{mb}^2 \rangle \le k^2 R_{mj}^2 \tag{272a}$$

$$\langle \delta R^T \delta R \rangle \le k^2 \hat{R}_j^T \hat{R}_j \tag{272b}$$

---

* It is pointed out that although the linear-combination start-up type of measurement appears at first sight to be a suitable candidate for exclusive use <u>all</u> the time, since it involves no measurement/state non-linearities at all, it has the serious disadvantage of requiring the simultaneous availability of data from more than one user/emitter link at a time. The basic linearized, one-emitter-at-a-time type of measurement is therefore to be preferred as the standard, with the linear combination type reserved for the nonlinear start-up situations.

** The conditions (246) are assumed.

*** The right sides of these equations are treated here as deterministic quantities.

Equations (272a and b) represent the desired tests for determining Kalman filter transition to and from start-up and standard operation. In particular, start-up operation (i.e., exclusive use of linear-combination measurements) should be initiated and should continue (with respect to a given net of emitters) until equations (272) are satisfied for <u>every</u> emitter j in the net. When all emitters satisfy (272), then standard measurement use should be initiated. In implementing the tests, the right sides of equations (272a) and (272b) should be computed from the quantities k, R_{mj}, and $\hat{R}_j$ as shown, while the left sides should be obtained from the Kalman filter covariance matrix P_K as follows. P_K can be written in partitioned form as.*

$$
P_K = \left[\begin{array}{c:c:c:c:c:c}
\langle \delta P \delta P^T \rangle & \langle \delta P \delta \dot{P}^T \rangle & \langle \delta P \delta R_{mb} \rangle & \langle \delta P \delta \dot{R}_{mb} \rangle & \langle \delta P \delta E^T \rangle & \langle \delta P \delta \dot{E}^T \rangle \\
\hdashline
\langle \delta \dot{P} \delta P^T \rangle & \langle \delta \dot{P} \delta \dot{P}^T \rangle & \langle \delta \dot{P} \delta R_{mb} \rangle & \langle \delta \dot{P} \delta \dot{R}_{mb} \rangle & \langle \delta \dot{P} \delta E^T \rangle & \langle \delta \dot{P} \delta \dot{E}^T \rangle \\
\hdashline
\langle \delta R_{mb} \delta P^T \rangle & \langle \delta R_{mb} \delta \dot{P}^T \rangle & \langle \delta R_{mb}^2 \rangle & \langle \delta R_{mb} \delta \dot{R}_{mb} \rangle & \langle \delta R_{mb} \delta E^T \rangle & \langle \delta R_{mb} \delta \dot{E}^T \rangle \\
\hdashline
\langle \delta \dot{R}_{mb} \delta P^T \rangle & \langle \delta \dot{R}_{mb} \delta \dot{P}^T \rangle & \langle \delta \dot{R}_{mb} \delta R_{mb} \rangle & \langle \delta \dot{R}_{mb}^2 \rangle & \langle \delta \dot{R}_{mb} \delta E^T \rangle & \langle \delta \dot{R}_{mb} \delta \dot{E}^T \rangle \\
\hdashline
\langle \delta E \delta P^T \rangle & \langle \delta E \delta \dot{P}^T \rangle & \langle \delta E \delta R_{mb} \rangle & \langle \delta E \delta \dot{R}_{mb} \rangle & \langle \delta E \delta E^T \rangle & \langle \delta E \delta \dot{E}^T \rangle \\
\hdashline
\langle \delta \dot{E} \delta P^T \rangle & \langle \delta \dot{E} \delta \dot{P}^T \rangle & \langle \delta \dot{E} \delta R_{mb} \rangle & \langle \delta \dot{E} \delta \dot{R}_{mb} \rangle & \langle \delta \dot{E} \delta E^T \rangle & \langle \delta \dot{E} \delta \dot{E}^T \rangle
\end{array}\right] \tag{273}
$$

$\langle \delta R_{mb}^2 \rangle$ is then available where shown in P_K.

*The state vector x and covariance matrix P_K represented in this appendix are (for brevity) actually only the subsets of elements of the more general x and P_K (as defined in AFAL-TR-72-80) which are involved in the measurement operations discussed here. Generalization of these results to apply to the more general x and P_K is simply a matter of inserting the necessary null relationships between the measurements defined here and the remaining, omitted state vector elements, and then reordering.

Since $<\delta R^T \delta R> = <\delta P^T \delta P + \delta E^T \delta E - 2\delta P^T \delta E> = <\delta P^T \delta P> + <\delta E^T \delta E> - 2<\delta P^T \delta E>$, it follows that $<\delta R^T \delta R>$ can then be obtained from P_K as:

$$<\delta R^T \delta R> = \mathrm{Tr}\ <\delta P \delta P^T> + \mathrm{Tr}\ <\delta E \delta E^T> - 2\mathrm{Tr}\ <\delta P \delta E^T> \quad (274)$$

where the symbol Tr denotes the trace (i.e., the sum of the diagonal elements) of the matrix following this symbol.

Finally, the constant k of equations (272) should be chosen small compared to unity (e.g., k = 0.1). However, simulation may be necessary to determine its most appropriate handling.

In navigation start-up with no a priori position or velocity information, $\hat{P}$, $\hat{\dot{P}}$, $\hat{E}$, $\hat{\dot{E}}$, $\hat{x}$, and P_K should be initialized according to

a) $\hat{P} = \hat{\dot{P}} = \hat{E} = \hat{\dot{E}} = \Delta\hat{R}^*_{mb} = \Delta\hat{\dot{R}}^*_{mb} = \hat{x} = 0$;

b) $<\delta P \delta P^T> = <\delta E \delta E^T> = <\delta P \delta E^T> = \sigma_p^2 I$ (where σ_p = large (1σ) position error);

$<\delta R_{mb}^2> = \sigma_R^2$, $<\delta \dot{R}_{mb}^2> = \sigma_{\dot{R}}^2$ (σ_R = 0.577 x lane width, $\dot{\sigma}_R$ = 0.577 x groundspeed capability of aircraft);

c) All other P_K elements = 0.

On the other hand, in E to EF frame transition, $<\delta E \delta E^T> = \sigma_E^2 I$, where σ_E = 1σ emitter net datum plane error.

---

* These are the a priori estimates of phase and phase rate error respectively which are subsequently corrected by the Kalman filter, and should not be confused with the corresponding Kalman filter estimates of errors in phase and phase rate.

APPENDIX X

IMU COARSE SELF-LEVELING AND ALIGNMENT

The algorithms developed in this appendix are based on two principal assumptions:

(a) The carrier vehicle either is stationary or is moving at constant speed and altitude on a great circle course throughout the entire coarse leveling and alignment operation.*

(b) Continuous radio-autonomous navigation has been established before the start of, and is maintained throughout the latter. coarse align phase. Also, the fixed transformation $T_{C/E}$ has been established

To begin, if v is the vehicle velocity with respect to the earth (E frame), then assumption (a) implies that:

$$\frac{d_L v}{dt} = 0 \tag{275}$$

i.e., the time rate of change of v with respect to the local vertical wander azimuth frame L is zero. It follows from (275) that:**

$$f = - g - \left(2\omega_{E/I} + \omega_{L/E}\right) \times v \tag{276}$$

Now f_{ACC}, the accelerometer output vector, is a direct measure of the specific force f. Neglecting the small last term, (276) can therefore be written:

$$f_{ACC} \approx -g \tag{277}$$

*These conditions need not and cannot be exactly satisfied; however, the inaccuracies in the coarse alignment scheme described here will be directly proportional to the deviations from them.

**See Appendix II, equation (46).

Since $-g$ has the direction of the local (upward) vertical, it follows that if ℓ_1 is the unit local vertical vector, then:

$$\ell_1 \approx f_{ACC} / \left| f_{ACC} \right| \tag{278}$$

The vector ℓ_1 provides the basis for platform coarse leveling. In particular:

$$\begin{gathered} \left(p_1\right)_P^T \left(\ell_1\right)_P \approx \cos\theta \\ \left(p_1\right)_P \times \left(\ell_1\right)_P = \left(u_1\right)_P \approx \sin\theta \frac{\left(u_1\right)_P}{\left|u_1\right|} \end{gathered} \tag{279}$$

where $\left(p_1\right)_P$ is the unit #1 P frame axis vector $\left\{ \left(p_1\right)_P^T = [\ 1\ 0\ 0] \right\}$ and θ ($0° \le \theta \le 180°$) is the angle between the platform #1 (azimuth) axis and the local vertical. The vector u_1 lies along the intersection of the local horizon plane and the plane defined by p_2 and p_3, and has a magnitude $\sin\theta$.

A very fast way to erect the platform would consist in applying slew rate along the platform direction defined by u_1; i.e.:

$$\left(\omega_{SLEW}\right)_P = \left|\omega_{SLEW}\right| \frac{\left(u_1\right)_P}{\left|u_1\right|} \tag{280}$$

The slew should be continued until:

$$\left|u_1\right| \le \theta_1 \text{ and } \left(p_1\right)_P^T \left(\ell_1\right)_P > 0 \tag{281}$$

where θ_1 is a suitable chosen constant.*

However, if only one level of fixed slew rate can be applied to each platform gyro, then (280) is not possible, and must be replaced by:

* This algorithm will probably work as it stands even when the platform is initially upside down. This should, however, be verified by simulation.

$$\left(\omega_{SLEW}\right)_P = \left|\omega_{SLEW}\right| \left(P_i\right)_P \tag{282}$$

until $\left(P_j\right)_P^T \left(\ell_1\right)_P < \theta_2$ (i = 2 or 3)

and then

$$\left(\omega_{SLEW}\right)_P = \left|\omega_{SLEW}\right| \left(P_j\right)_P \tag{283}$$

until $\left(P_i\right)_P^T \left(\ell_1\right)_P < \theta_2$ $(j \neq i)$

where θ_2 is another suitably chosen constant.

When coarse erection is complete, a period of proportional control leveling should follow in order to attain a sufficiently accurate platform vertical to allow subsequent coarse alignment. This can be done by applying the gyro torquing rate:

$$\left(\omega_{PROPL}\right)_P = k_{PROPL} \left(u_1\right)_P \tag{284}$$

where k_{PROPL} is an appropriate proportional leveling control gain. In particular, this rate should be maintained until the end of coarse alignment.

When:

$$\left|u_1\right| < \theta_3 \tag{285}$$

where θ_3 is an appropriate constant, the null rate applied to the azimuth gyro* should be replaced by the azimuth earth rate component:

$$\left(\omega_{PROP1}\right)_P = \left\{\left(\omega_{E/I}\right)_C^T \frac{g_c}{|g|}\right\} \left(P_1\right)_P \tag{286}$$

where g_c is available as a dynamic VSM output.

---

* This null rate is implied by equation 10), since $\left(u_1\right)_P$ is a 3 X 1 vector whose #1 element is zero.

The total stabilization rate:

$$\left(\omega_{GYR}\right)_P = \left(\omega_{PROPL}\right)_P + \left(\omega_{PROP1}\right)_P \tag{287}$$

should continue to be applied to the platform gyros until coarse align is complete.

With the platform held level and approximately nonrotating in azimuth (with respect to a wander azimuth frame) by the torquing rate (287), the coarse align phase can begin.

Since, except for platform drift rates and the azimuth earth rate misalignment produced by the small platform hangoff, the applied rate (287) is equal to the sum of local earth rate and the local vertical angular rate, it follows that:

$$\left(\omega_{GYR}\right)_P \approx T_{P/C}\left(\omega_{P/I}\right)_C \tag{288}$$

where:

$$\left(\omega_{P/I}\right)_C = \left(\omega_{P/C}\right)_C + \left(\omega_{E/I}\right)_C \tag{289}$$

Also:

$$\left(f_{ACC}\right)_P \approx -T_{P/C}g_C \tag{290}$$

and taking the vector cross-product of equations (288) and (290) gives:

$$\left(f_{ACC}\right)_P \times \left(\omega_{GYR}\right)_P \approx T_{P/C}\left[\left(\omega_{P/I}\right)_C \times g_C\right] \tag{291}$$

Equations (288), (290), and (291) can be combined into a single 3x3 matrix equation, and solved explicitly for $T_{P/C}$ as*:

*An alternate, normalized form of (292) can be obtained by first normalizing equations (288), (290) and (291). This form should also be considered when it comes to actual programming for a specific application.

$$T_{P/C} = \left[\left(f_{ACC}\right)_P \,\vdots\, \left(\omega_{GYR}\right)_P \,\vdots\, \left(f_{ACC}\right)_P \times \left(\omega_{GYR}\right)_P\right] \left[-g_C \,\vdots\, \left(\omega_{P/I}\right)_C \,\vdots\, \left(\omega_{P/I}\right)_C \times g_C\right]^{-1} \quad (292)$$

In equation (292), g_C and $\left(\omega_{E/I}\right)_C$ are available directly as VSTM outputs. $\left(\omega_{P/C}\right)_C$ can also be computed from VSTM outputs as follows.

Now:

$$\left(\omega_{P/C}\right)_C = T_{C/P}\left(\omega_{P/C}\right)_P$$

But, if L is any locally level, wander azimuth frame, then because of assumption (a) above, it follows that $\omega_{P/C} = \omega_{L/C}$. Therefore:

$$\begin{aligned}
\left(\omega_{P/C}\right)_C &= T_{C/P}\left(\omega_{L/C}\right)_P \\
&= T_{C/P}T_{P/L}\left(\omega_{L/C}\right)_L \\
&= T_{C/L}\left(\omega_{L/C}\right)_L
\end{aligned}$$

In the L frame, however, $\omega_{L/C}$ can be expressed in terms of v_L or v_C as:

$$\left(\omega_{L/C}\right)_L = \frac{1}{R_o} K_{Lo} v_L = \frac{1}{R_o} K_{Lo} T_{L/C} v_C$$

where R_o is the nominal radius of the earth and:

$$K_{Lo} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} = \frac{-g_L}{|g_L|} \times$$

Therefore:

$$\left(\omega_{P/C}\right)_C = \frac{1}{R_o}\left(T_{C/L} K_{Lo} T_{L/C}\right) v_C$$

or

$$\left(\omega_{P/C}\right)_C = \frac{1}{R_o} K_{Co} v_C$$

where:

$$K_{Co} = \frac{-g_C}{|g|} \times$$

$\left(\omega_{P/C}\right)_C$ can therefore be determined using g_C and v_C from the VSTM, by the formula:

$$\left(\omega_{P/C}\right)_C = \frac{1}{R_o} \frac{-g_C}{|g|} \times v_C \tag{293}$$

Equation (292) reflects the impossibility of stationary gyrocompassing (i.e., azimuth determination) in the vicinity of the earth's poles in the fact that $\omega_{P/I} \times g = 0$ (since $\omega_{P/I}$ and g are parallel) in this case, so that the indicated matrix inversion required to determine $T_{C/P}$ is impossible. Also, if the carrier vehicle is moving in such a way as to remain essentially nonrotating with respect to inertial space, then $\omega_{P/I\ P} \approx 0$, and gyrocompassing is again impossible*.

However, for all other types of vehicle motion, gyrocompassing is possible. In particular, it <u>is</u> possible for a moving vehicle at the pole, provided that sufficient radio data is available to continuously maintain accurate VSTM p_C and v_C outputs during the align phase.

When only an AHRU is available as an attitude reference, then $T_{P/C}$ cannot be initialized directly as above. Rather $T_{L/C}$ and $T_{P/L}$ must first be initialized and then $T_{P/C}$ computed from:

$$T_{P/C} = T_{P/L} T_{L/C}$$

*Mathematically, this condition is described by: $v_C = -T_{C/E}\left\{\left(\omega_{E/I}\right)_E \times p_E\right\}$. Strictly speaking, this situation can only arise on small-circle, not great-circle, courses. However, it can be closely approached at the high-latitude apogee of certain great-circle courses.

$T_{L/C}$ for this purpose can be obtained as:

$$T_{L/C} = \begin{bmatrix} \left(L_1\right)_C^T \\ \left(L_2\right)_C^T \\ \left(L_3\right)_C^T \end{bmatrix}$$

$$\left(L_1\right)_C = -g_C/|g| \qquad \text{(unit vertical up)}$$

$$\left(L_2\right)_C = \frac{\left(\omega_{P/I}\right)_C \times \left(L_1\right)_C}{|\left(\omega_{P/I}\right)_C \times \left(L_1\right)_C|}$$

$$\left(L_3\right)_C = \left(L_1\right)_C \times \left(L_2\right)_C$$

where L_2, L_3 = unit east, north vectors if $v_C = 0$
= unit along-track, cross-track vectors at earth's poles

In general, if L' is any other locally level frame (i.e., such that $L_1' = L_1$) then the transformation from the L to the L' frames can be represented by

$$T_{L'/L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{L'/L} & -\sin\theta_{L'/L} \\ 0 & \sin\theta_{L'/L} & \cos\theta_{L'/L} \end{bmatrix}$$

where $\theta_{L'/L}$ is the CCW angle from L_3 to L_3'.

If L' = local up, east, north frame, then:

$$\begin{cases} \sin\theta_{L'/L} = \left(L_2\right)_C^T n_C \\ \cos\theta_{L'/L} = \left(L_2\right)_C^T e_C \end{cases}$$

where: $\begin{cases} e_C = \text{unit east vector} = \left(L_2\right)_C, \text{ with } v_C \text{ set to zero} \\ n_C = \text{unit north vector} = \left(L_1\right)_C \times e_C \end{cases}$

Near the poles, where L' cannot be defined in this way since north is undefined, $\theta_{L'/L}$ can be defined by the more general equations:

$$\begin{cases} \sin \theta_{L'/L} = \left(L_2\right)_C^T \left(L_{2C}\right)' \\ \cos \theta_{L'/L} = \left(L_2\right)_C^T \left(L_{3C}\right)' \end{cases}$$

where $\left(L_2\right)_C'$ must be defined with respect to the earth-fixed C frame, and $\left(L_{3C}\right)' = \left(L_1\right)_C \times \left(L_{2C}\right)'$

APPENDIX XI

ANTENNA LEVER ARM ERRORS

This appendix contains the development for establishing the errors due to uncertainty in aircraft frame direction cosines and gives a means of reducing the state complexity and redundancy contained in Appendix VI.

A basic linear error effect is the uncertainty in defining the antenna lever arm of the user. Note that the error vector is defined by assuming the following conditions:

(a) Antenna separation from platform is fixed and invariant in the airframe coordinate system.

(b) A direction cosine matrix (DCM) is defined which transforms from aircraft coordinate system to computer-defined earth-centered system.

Notationally the lever arm is defined as:

$$d_C = T_{C/A} d_A \tag{294}$$

where d_A = antenna lever arm in aircraft frame, vector

d_C = antenna lever arm in computer frame, vector

$T_{C/A}$ = direction cosine matrix (DCM) from aircraft to computer axes.

In order to establish the error vector for the antenna lever arm, consider the following:

$$d + \Delta d = \left|T_{C/A} + \delta T_{C/A}\right|\left(d_A + \Delta d_A\right) \tag{295}$$

$$d + \Delta d = T_{C/A} d_A + T_{C/A} \Delta d_A + \delta T_{C/A} d_A + \delta T_{C/A} \Delta d_A \tag{296}$$

and

$$\Delta d = T_{C/A} \Delta d_A + \delta T_{C/A} \Delta d_A + \delta T_{C/A} d_A \tag{297}$$

where $\delta T_{C/A}$ = 3x3 direction cosine matrix which is functional with $T_{C/A}$ and with the errors in establishing $T_{C/A}$

Δd_A = error in antenna lever arm as defined along the aircraft axis.

The DCM $\delta T_{C/A}$ is also given by:

$$T_{C/A} + \delta T_{C/A} = \left|I + \delta T_{C/A} T_{A/C}\right|\left|T_{C/A}\right| \tag{298}$$

where

$\delta T_{C/A} T_{A/C}$ = skew symmetric matrix which for small angle errors is the vector matrix $[\phi x]$

$$\delta T_{C/A} = [\phi] T_{C/A} = \begin{bmatrix} 0 & \phi_z & -\phi_y \\ -\phi_z & 0 & \phi_x \\ \phi_y & -\phi_x & 0 \end{bmatrix} T_{C/A} \tag{299}$$

The proof of the logic given for defining the Incremental Direction Cosine Matrix $\delta T_{C/A}$ is given by the following:

Consider the DCM:

$$T + \delta T$$

We can write

$$T + \delta T = \left| I + \delta T T^T \right| T \tag{300}$$

since $\delta T T^T T = \delta T [I]$

$$T + \delta T = T + \delta T \left| T^T T \right|$$

$$= \left| I + [\phi x] \right| T \tag{302}$$

so $[\phi x] = \delta T T^T$ (303)

$$[\phi x] T = \delta T T^T T = \delta T\ [I] = \delta T. \tag{304}$$

where $[\phi x]$ = skew symmetric matrix.

Note that the formulation of

$$\delta T_{C/A} = [\phi x] T_{C/A} \tag{305}$$

requires only the definition of a skew symmetric matrix of 3 unknown vector misalignment angles:

$$\phi = \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix}$$

rather than the 9 elements of $\delta T_{C/A}$.

The approach defined for all 9 elements is:

$$\Delta d = \underbrace{\left[T_{C/A} + \delta T_{C/A}\right]\Delta d_A}_{\substack{\text{Observable portion}\\ \text{of any error of}\\ \text{antenna location}\\ \downarrow\\ \text{or keep in state vector}\\ \text{for the filter to estimate.}}} + \underbrace{\delta T_{C/A} d_A}_{\text{Assume} = 0} \tag{307}$$

The latter portion may be written as:

$$\delta T_{C/A} = \begin{bmatrix} 0 & \phi_z & -\phi_y \\ -\phi_z & 0 & \phi_x \\ \phi_y & -\phi_x & 0 \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix},$$

$$= \begin{matrix} T_{21}\phi_z - T_{31}\phi_y & T_{22}\phi_z - T_{32}\phi_y & T_{23}\phi_z - T_{33}\phi_y \\ -T_{11}\phi_z + T_{31}\phi_x & -T_{12}\phi_z + T_{32}\phi_x & -T_{13}\phi_z + T_{33}\phi_x \\ T_{11}\phi_y - T_{21}\phi_x & T_{12}\phi_y - T_{22}\phi_x & T_{13}\phi_y - T_{23}\phi_x \end{matrix}$$

$$= 3\text{x}3 = \begin{matrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{matrix} \tag{308}$$

and $\left[\delta T_{C/A}\right] d_A = [3\text{x}3] \begin{bmatrix} dA_1 \\ dA_2 \\ dA_3 \end{bmatrix}$

$$= \begin{matrix} \phi_{11} dA_1 + \phi_{12} dA_2 + \phi_{13} dA_3 \\ \phi_{21} dA_1 + \phi_{22} dA_2 + \phi_{23} dA_3 \\ \phi_{31} dA_1 + \phi_{32} dA_2 + \phi_{33} dA_3 \end{matrix} = 3\text{x}1 \text{ vector} \tag{309}$$

Expanding the terms for the indicated product yields:

$$\Delta d_1 = \left(T_{21}\phi_z - T_{31}\phi_y\right)dA_1$$
$$\left(T_{22}\phi_z - T_{32}\phi_y\right)dA_2$$
$$\left(T_{23}\phi_z - T_{33}\phi_y\right)dA_3 \tag{310}$$

$$\Delta d_2 = \left(-T_{11}\phi_z + T_{31}\phi_x\right)dA_1$$
$$\left(-T_{12}\phi_z + T_{32}\phi_x\right)dA_2$$
$$\left(-T_{13}\phi_z + T_{33}\phi_x\right)dA_3 \tag{311}$$

$$\Delta d_3 = \left(T_{11}\phi_y - T_{21}\phi_x\right)dA_1$$
$$\left(T_{12}\phi_y - T_{22}\phi_x\right)dA_2$$
$$\left(T_{13}\phi_y - T_{23}\phi_x\right)dA_3 \tag{312}$$

What is wanted is the following:

$$\begin{bmatrix} \Delta d_1 \\ \Delta d_2 \\ \Delta d_3 \end{bmatrix} = \begin{bmatrix} M \end{bmatrix} \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} \tag{313}$$

The elements of [M] can be defined as:

$$M_{11} \triangleq 0.0$$
$$M_{22} \triangleq 0.0$$
$$M_{33} \triangleq 0.0 \tag{314}$$

$$M_{12} = -T_{31}dA_1 - T_{32}dA_2 - T_{33}dA_3$$

$$= - \begin{bmatrix} T_{31} & T_{32} & T_{33} \end{bmatrix} \begin{bmatrix} dA_1 \\ dA_2 \\ dA_3 \end{bmatrix} \tag{315}$$

$$M_{13} = T_{21}dA_1 + T_{22}dA_2 + T_{23}dA_3$$

$$= \begin{bmatrix} T_{21} & T_{22} & T_{23} \end{bmatrix} \begin{bmatrix} dA_1 \\ dA_2 \\ dA_3 \end{bmatrix} \tag{316}$$

$$M_{21} = \begin{bmatrix} T_{31} & T_{32} & T_{33} \end{bmatrix} \begin{bmatrix} dA_1 \\ dA_2 \\ dA_3 \end{bmatrix} \tag{317}$$

$$M_{23} = - \begin{bmatrix} T_{11} & T_{12} & T_{13} \end{bmatrix} \begin{bmatrix} dA_1 \\ dA_2 \\ dA_3 \end{bmatrix} \tag{318}$$

and $$M_{31} = - \begin{bmatrix} T_{21} & T_{22} & T_{23} \end{bmatrix} \begin{bmatrix} dA_1 \\ dA_2 \\ dA_3 \end{bmatrix} \tag{319}$$

$$M_{32} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \end{bmatrix} \begin{bmatrix} dA_1 \\ dA_2 \\ dA_3 \end{bmatrix} \tag{320}$$

which defines $[M]$ to be a <u>skew symmetric</u> matrix with 6 nonzero elements, 3 of which are identical except for sign.

Note that each element of M is basically a dot product of two vectors which are:

$$T = \begin{bmatrix} T_1^T \\ \hdashline T_2^T \\ \hdashline T_3^T \end{bmatrix} \qquad \text{and } T^T = \left[T_1 \;\vdots\; T_2 \;\vdots\; T_3 \right] \tag{321}$$

$$\begin{aligned} \text{and } T_1 &= \text{column vector of } T^T \\ T_2 &= \text{column vector of } T^T \\ T_3 &= \text{column vector of } T^T \end{aligned} \tag{322}$$

$$\text{and } M = \begin{bmatrix} 0 & -T_3 \cdot dA & \overline{T}_2 \cdot dA \\ \overline{T}_3 \cdot dA & 0 & -T_1 \cdot dA \\ -T_2 \cdot dA & T_1 \cdot dA & 0 \end{bmatrix} \tag{323}$$

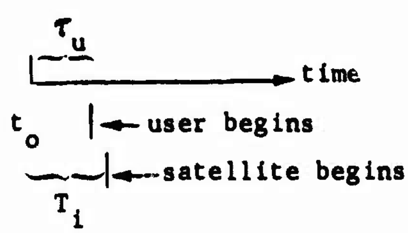
APPENDIX XII

LOS RANGE AND RANGE RATE TIME DELAYS PERTINENT TO MULTILATERATION

This appendix is concerned with the mechanization and timing sequence due to transit and computational delays for a pseudorange and range-rate system. The development is intended to answer some bothersome aspects of the timing and control sequence. In general, this appendix can also be considered in part an extension of the material and results of Appendix VI.

1. PSEUDONOISE RANGING CONCEPTS FOR ERROR ESTIMATION

Consider a universal time track defined as t_o:



At $t_o+\tau_u$ a user receiver generates a pseudorandom noise phase-shift keyed code sequence which is used to cross-correlate with a signal generated by the ith satellite which also generates the same PRN PSK sequence, but with a transmission incidence of starting given as $t_o+\tau_i$.

Note that in both cases the universal clock delays can be defined as range biases given by:

$$b_u = c\tau_u \qquad c = \text{speed of light}$$

$$b_i = c\tau_i$$

In the same universal time track we define that we have knowledge of the satellite position vector or

$$e_i(t_o) \triangleq \text{known satellite position vector}$$

Actually all we will have in reality will be an estimate of the satellite vector or

$$\hat{e}(t_o) = e(t_o) + \Delta d(t_o) \tag{324}$$

The position of the user is given by a p vector or

$$p(t_o) \triangleq \text{defined user position at } t_o$$

The exact range between the user and the satellite is defined as the scalar quantity (see Figure 38).

$$|R| = |p-e| \tag{325}$$

or $$|R(t_o)| = |p(t_o) - e(t_o)|$$

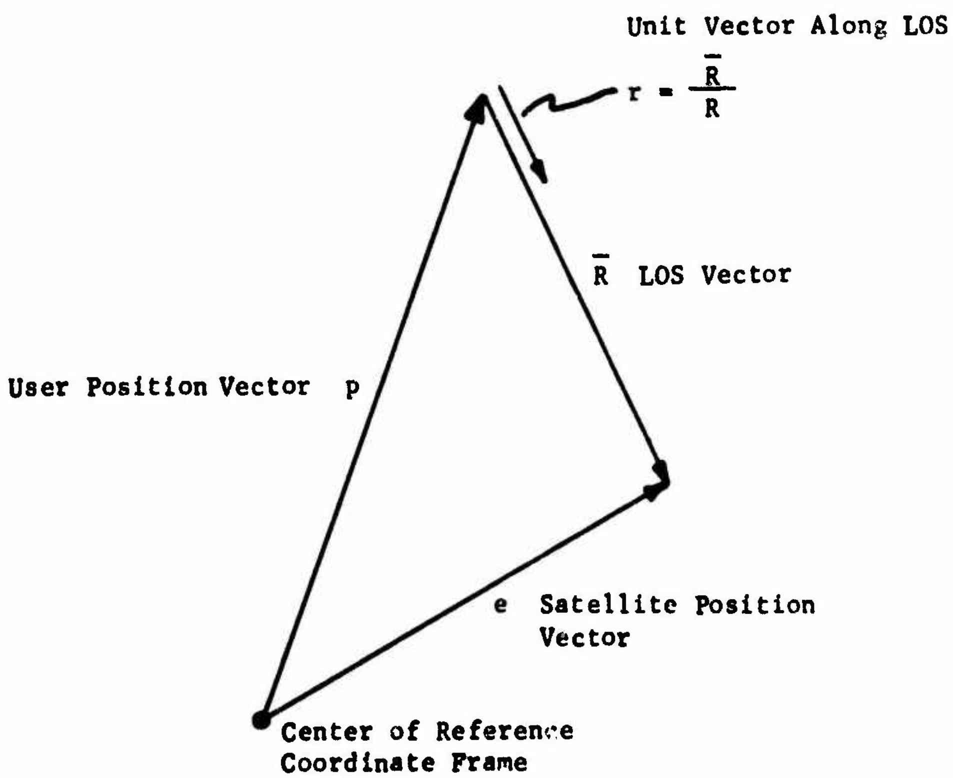


Figure 38. User-Receiver Vector Geometry

It is this range which causes a time delay of the code sequence measured at the receiver so that the receiver cross-correlation measures:

$$T_j = \frac{|R(t_o)|}{c} + \Delta T_d + \delta T - \tau_u + \tau_i + \Delta T_L$$

where ΔT_d = time delay through receiver

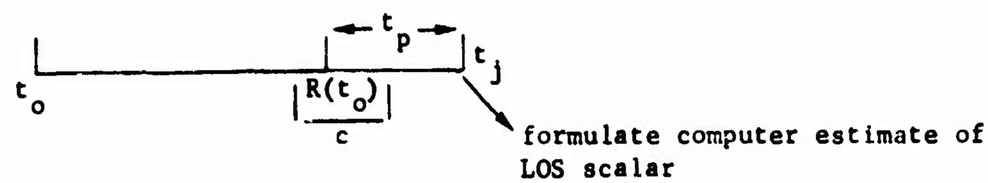
δT = noise in measurement

ΔT_L = link delays due to troposphere and ionosphere.

This measured time delay is scaled into range by

$$cT_j = |R(t_o)| + c\Delta T_d + c\delta T - b_u + b_i + c\Delta T_L \tag{326}$$

This range measurement is not obtained with respect to universal time until t_j where t_j is the sum of at least the delay time $\frac{|R(t_o)|}{c}$ plus a processing delay t_p.



Assume that at time t_j' in the receiver-computer, which is $t_j - \tau_u$, the receiver transfers the measurement T_j to the computer and the computer forms the following estimate of scalar LOS range:

$$|R(t_j') + \Delta R(t_j')| = |\hat{p}(t_j') - \hat{e}(t_j')| \tag{327}$$

where $\hat{p}(t_j') = p(t_j') + \Delta p(t_j')$ (true user position plus error)

$\hat{e}(t_j) = e(t_j') + \Delta e(t_j')$ (true emitter position plus error)

but $p(t_j') = p(t_o) + \dot{p}(t_o)\ (t_j' - t_o)$*

and $\Delta p(t_j') = \Delta p(t_o) + \Delta\dot{p}(t_o)\ (t_j' - t_o)$

*This is of course an approximation, since acceleration and higher order terms have been neglected.

So $\quad p(t_j') = p(t_o) + \dot{p}(t_o)\,(t_i - \tau_u - t_o)$

and similarly for emitter position. But we know $t_j - t_o = T_j + t_p$; thus the term provided by the computer will be:

$$\left|p(t_o) + \dot{p}(t_o)\,(T_j + t_p - \tau_u) - e(t_o) - \dot{e}(t_o)\,(T_j + t_p - \tau_u)\right|$$
$$+ \left|\Delta p(t_o) + \Delta\dot{p}(t_o)\,(T_j + t_p - \tau_u) - \Delta e(t_o) - \Delta\dot{e}(t_o)(T_j + t_p - \tau_u)\right|$$

The observable provided to the Kalman filter will be the following difference:

$$Y = \text{observable error}$$

$$Y = \left|R(t_j') + \Delta R(t_j')\right| - cT_j \tag{328}$$

Let r = unit vector defined as $\frac{\bar{R}}{|R|}$ along the LOS; then we may write the above equation as the following dot or transpose product:

$$\begin{aligned} Y = {} & r^T p\,(t_o) + r^T\dot{p}(t_o)(T_j + t_p - \tau_u) \\ & -r^T e(t_o) - r^T\dot{e}(t_o)(T_j + t_p - \tau_u) \\ & +r^T\Delta p(t_o) + r^T\Delta\dot{p}(t_o)(T_j + t_p - \tau_u) \\ & -r^T\Delta e(t_o) - r^T\Delta\dot{e}(t_o)(T_j + t_p - \tau_u) \\ & -r^T p(t_o) + r^T e(t_o) - c\Delta T_d - c\delta T + b_u - b_i - c\Delta T_L \end{aligned} \tag{329}$$

Canceling out the common terms of the above, we obtain:

$$\begin{aligned} Y = {} & \text{function of error states } (\Delta p, \Delta e, \Delta\dot{p}, \Delta\dot{e}) \\ & + \text{function of timing error } (b_u, b_i, c\Delta t_d, c\Delta t_L) \\ & + \text{measurement noise } (c\delta T) \end{aligned}$$

If we define the state vector for our system as the following $\underline{n}$ element vector, the entire observable equation may be expressed as:

$$\bar{Y} = Mx + \bar{v} \tag{330}$$

where $$x = \begin{vmatrix} \Delta p \\ \Delta\dot{p} \\ \Delta e \\ \Delta\dot{e} \\ b_u \\ b_i \\ c\Delta T_L \\ t_p \\ c\Delta T_d \end{vmatrix}$$

Substituting for T_j, we obtain:

$$(T_j + t_p - \tau_u) = \begin{bmatrix} \frac{r^T}{c}\, p(t_o) - \frac{r^T}{c}\, e(T_o) + \Delta T_d + t_p - 2\tau_u + \tau_i + \Delta T_L \\ + \delta T \quad \text{random measurement noise} \end{bmatrix}$$

$$\left.\begin{aligned} &\text{So}\quad r^T\dot{p}(t_o) + r^T\Delta\dot{p}(t_o) = r^T\hat{\dot{p}}(t_o) \\ &\text{and}\quad r^T\dot{e}(t_o) + r^T\Delta\dot{e}(t_o) = r^T\hat{\dot{e}}(t_o) \end{aligned}\right\} \quad \text{values of estimated velocity, uncontrolled}$$

$$\bar{v} = -c\delta T$$

Examination of equation (329) indicates that using nonsynchronous data differences leads to the additional terms in the measurement (relative to those required for synchronous differences; see Appendix VI):

$$\Delta Y = r^T\hat{\dot{p}}\,(T_j + t_p - \tau_u) - r^T\hat{\dot{e}}\,(T_j + t_p - \tau_u) \tag{331}$$

These terms can be compensated, since we do indeed have knowledge of $\hat{\dot{p}}(t_j')$ and $\hat{\dot{e}}(t_j)$ and T_j. Therefore a compensating $\Delta\hat{Y}$ can be generated such that:

$$\begin{aligned} \Delta\hat{Y} = &- r^T\hat{\dot{p}}(t_j' - T_j)\ \left|T_j\right| \\ &+ r^T\hat{\dot{e}}(t_j' - T_j)\ \left|T_j\right| \end{aligned} \tag{332}$$

Such a mechanization would require a backward propagation in time to the approximate time t'_o which is in error from true time, t_o, by $(t_p - \tau_u)$.

It would also of course be possible to compensate the measured time delay t_j itself, instead of the range, to obtain the approximate delay time, but this is conceptually the same as the above and hence should make no difference in terms of the error coupling. The observable ΔY can be written after compensation as:

$$\begin{aligned}\Delta Y = & \left[r^T\dot{p}(t_o) - r^T\dot{p}(t_o')\right]\left[T_j\right] \\ & + r^T\dot{p}(t_o)\left[t_p - \tau_u\right] \\ & - \left[r^T\dot{e}(t_o) - r^T\dot{e}(t_o')\right]\left[T_j\right] \\ & - r^T\dot{e}(t_o)\left[t_p - \tau_u\right]\end{aligned} \tag{333}$$

At first glance this may not appear to be much of a simplification, but if we assume that the acceleration is small or zero, then:

$$\begin{aligned}\Delta Y \cong & \; r^T\dot{p}(t_o)\left[t_p - \tau_u\right] \\ & - r^T\dot{e}(t_o)\left[t_p - \tau_u\right]\end{aligned} \tag{334}$$

Hence the terms of the observable with velocity compensation are:

$$\begin{aligned}Y = & \; r^T\Delta p(t_o) - r^T\Delta e(t_o) + r^T\dot{p}(t_o)t_p - r^T\dot{e}(t_o)t_p \\ & - \left[r^T\dot{p}(t_o) - c\right]\tau_u + r^T\dot{e}(t_o)\tau_u \\ & - c\Delta T_d - c\Delta T_L - c\tau_i \\ & + c\delta T\end{aligned} \tag{335}$$

Let M^T = transpose of the measurement matrix; then:

$$M^T = \begin{array}{l} r^T \\ 0 \\ -r^T \\ 0 \\ 1 - \dfrac{r^T \dot{p}(t_o)}{c} + \dfrac{r^T \dot{e}(t_o)}{c} \\ -1 \\ -1 \\ r^T \dot{p}(t_o) - r^T \dot{e}(t_o) \\ -1 \end{array} \qquad (336)$$

Note that since the propagation-delay-dependent elements of M involve a division by the speed of light, these are extremely small terms. Hence, with a velocity-compensated mechanization which has <u>no</u> fixed processing delay time, t_p, the measurement matrix reduces to that for the short-range LOS case of synchronous, almost instantaneous (because short-range) propagation.

Table LXXXIV summarizes, for convenience of reference, the event timing involved in the foregoing discussion.

TABLE LXXXIV. SUMMARY OF EVENT TIMING

Universal
Time
Index

$|$———————————$|$
t_o $\qquad\qquad\qquad\qquad$ t_j

$\underbrace{\quad}_{\tau_u}|$ user clock delay with respect to universal time

$\underbrace{\quad}_{\tau_i}|$ satellite clock delay with respect to universal time

$|\longleftarrow T_j \longrightarrow|$ measured delay time for range correlation

$\underbrace{|\quad|}_{t_p}$ processing delay for measurement

$|$ data transferred to computer

$|\longleftarrow T_j$ ——— $|$ transition back to approximate t_o

t_o' approximate zero time using user clock

2. OBSERVATION OF PSEUDORANGE RATE FOR ERROR ESTIMATION

For moving vehicles the measurement of the doppler shift frequency of the carrier is given by

$$f_D = \frac{2}{\lambda} \frac{v \cdot R}{|R|} \tag{337}$$

where $v = \dot{p} - \dot{e}$

$R = p - e$

λ = carrier frequency.

Given that the user and satellite clocks are varying with rates of

$\dot{b}_u$ = user oscillator rate in fps

$\dot{b}_i$ = satellite oscillator rate in fps

the measured doppler term or pseudorange rate is thus given as:

$$\frac{2}{\lambda} |\dot{R}| = \frac{2}{\lambda} \left[\frac{v \cdot R}{|R|} + \dot{b}_u + \dot{b}_i + \Delta\dot{L} \right] \tag{338}$$

where $\Delta\dot{L}$ = transmission link range rate.

Realistically any doppler measurement requires a finite amount of time to obtain the frequency measurement so that equation (338) becomes

$$|\dot{\bar{R}}| \approx \frac{\lambda}{2} \int_{t_{n-1}}^{t_n} \frac{f_D(t_{n-1})}{\Delta t} dt + \frac{\lambda}{2} \int_{t_{n-1}}^{t_n} \dot{f}_D(t) dt + \dot{b}_u + \dot{b}_i + \Delta\dot{L} \tag{339}$$

where $\dot{f}_D(t)$ = derivative of doppler shift

Δt = measurement interval = $t_n - t_{n-1}$

Since timing is defined by the user clock, the actual universal time definition is related to the above by $(t - T_n)$ where T_n = propagation delay time. Note also that the interval Δt has a timing error which is due to the user clock rate:

$$\Delta t' = \Delta t - \frac{\dot{b}\Delta t'}{c} \cong \Delta t \left(1 - \frac{\dot{b}}{c}\right) \tag{340}$$

In summary, any data measured at the receiver is delayed by the propagation time, and the data is averaged over an incorrect time interval due to the clock drift rate.

The nature of the frequency shift derivative can be formulated by the following:

$$\frac{d}{dt}(f_D) = \frac{d}{dt}\left\{\frac{2}{\lambda}[v\cdot R][|R|]^{-1}\right\}$$

$$= \frac{2}{\lambda}\left[\frac{d}{dt}[v\cdot R]\,[|R|]^{-1} + [v\cdot R][-1][|R|]^{-2}\frac{d}{dt}[|R|]\right]$$

$$= \frac{2}{\lambda}\left[\frac{d}{dt}\frac{[v\cdot R]}{|R|} - f_D\frac{d}{dt}\frac{[|R|]}{|R|}\right]$$

$$= \left[\frac{dv}{dt}\cdot(R) + (v)\cdot\frac{dR}{dt} - f_D\frac{d}{dt}[|R|]\right]\frac{2}{\lambda|R|} \tag{341}$$

Hence the derivative consists of three terms: two dot products and a derivative of the scalar range. These three parts of the derivative may be written as:

$$\frac{d}{dt}(f_D) = \frac{2}{\lambda}\left[\frac{dv}{dt}\cdot\frac{R}{|R|}\right]$$

$$+ \frac{2}{\lambda}\left[\frac{v}{|R|}\cdot\frac{dR}{dt}\right]$$

$$- f_D\left[R\cdot\frac{dR}{dt}\right]\frac{1}{|R|^2} \tag{342}$$

Obviously if the quantities $v(t)$ and $R(t)$ have significant time variations during the interval Δt, the formulation of $|\dot{R}|(t)$ could be rather complex. Therefore assume for the moment that Δt is sufficiently small so that equation (337) describes the range rate scalar relation defined within the computer; hence at time t_m the following values are generated:

$$\left.\begin{aligned} v(t_m) &= \hat{\dot{p}}(t_m) - \hat{\dot{e}}(t_m) \\ R(t_m) &= \hat{\bar{p}}(t_m) - \hat{\bar{e}}(t_m) \end{aligned}\right\} \quad \text{estimates of velocity and position}$$

The velocity and position estimates may be written as:

$$\hat{v}(t_m) = \dot{p}(t_m) + \Delta\dot{p}(t_m) - \dot{e}(t_m) - \Delta\dot{e}(t_m)$$

$$\hat{R}(t_m) = p(t_m) + \Delta p(t_m) - e(t_m) - \Delta e(t_m)$$

Using the results of equation (341), we may write the error in f_D, the doppler frequency, as:

$$\Delta f_D = \frac{2}{\lambda}\left[\frac{(\Delta\dot{p}-\Delta\dot{e})\cdot(p-e)}{\left(|p-e|\right)}\right] + \frac{2}{\lambda}\left[\frac{(\dot{p}-\dot{e})\cdot(\Delta p-\Delta e)}{\left(|p-e|\right)}\right] - f_D\left[\frac{(p-e)\cdot(\Delta p-\Delta e)}{\left(|p-e|^2\right)}\right]$$

Or, using the unit LCS direction cosine vector r,

$$\Delta f_D = \frac{2}{\lambda}\left[(\Delta\dot{p}-\Delta\dot{e})\,r^T\right] + \frac{2}{\lambda}\left[\frac{(\dot{p}-\dot{e})\cdot(\Delta p-\Delta e)}{|R|}\right] - \frac{f_D}{|R|}\left[(\Delta p-\Delta e)\,r^T\right] \tag{344}$$

The last two terms here can be written:

$$\frac{2}{\lambda}\left[\frac{(\dot{p}-\dot{e})^T(\Delta p-\Delta e)}{|R|} - \frac{(\dot{p}-\dot{e})^T}{|R|}\,r\,(\Delta p-\Delta e)\,r^T\right]$$

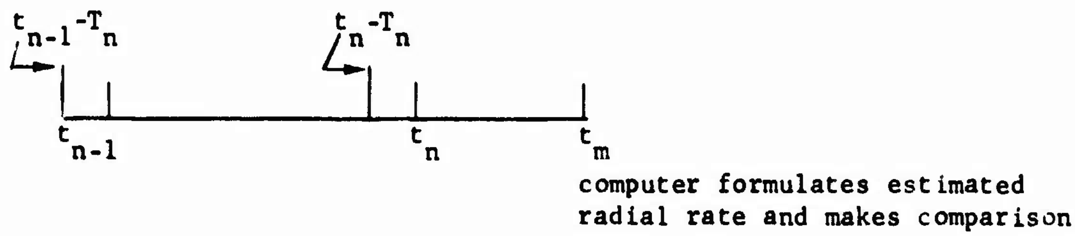
$$= \frac{2}{\lambda}\left[\frac{(\dot{p}-\dot{e})^T}{|R|}\left|I - rr^T\right|\right](\Delta p-\Delta e) \tag{345}$$

With this simplified form we now have a means of expressing measured observables of data as the difference between the computed range rate and the measured range rate.

Let Y = observable for the filter; then:

$$
\begin{aligned}
Y = \frac{2}{\lambda} \Bigg\{ & (p-e)\, r^T + (\Delta\dot{p} - \Delta\dot{e})\, r^T \\
& + \left[\frac{(\dot{p}-\dot{e})^T}{|R|} \left[I - rr^T \right] \right] (\Delta p - \Delta e) \\
& - \int_{t_{n-1}}^{t_n} f_D \frac{(t_{n-1})\, dt}{\Delta t} - \int_{t_{n-1}}^{t_n} f_D(t)\, dt - \dot{b}_u \frac{2}{\lambda} - \dot{b}_i \frac{2}{\lambda} - \Delta\dot{L} \frac{2}{y} \Bigg\}
\end{aligned}
\tag{346}
$$

In order to understand the time span for the above, consider that the universal time axis is given as:



For the sake of brevity and further understanding, consider that

$$
\frac{2}{\lambda} \left[(\dot{p}(t_m) - e(t_m)) \right] r^T - \int_{t_{n-1}}^{t_n} \frac{f_D(t_{n-1})}{\Delta t} - \int_{t_{n-1}}^{t_n} \dot{f}_D(t)\, dt
$$

is most likely nonzero and

$$
= \Delta\dot{B} \frac{2}{\lambda}
$$

where $\Delta\dot{B}$ = dynamic doppler lag due to nonsynchronous sampling and finite doppler bandwidth and delay.

And thus in <u>velocity</u> terms:

$$Y = \left[\Delta\dot{p}(t_m) - \Delta\dot{e}(t_m)\right] r^T$$
$$+ \left[\frac{\left[\dot{p}(t_m) - \dot{e}(t_m)\right]^T}{|R|} \left[I - rr^T\right] \right] \left[\Delta p(t_m) - \Delta e(t_m)\right]$$
$$- \dot{b}_u - \dot{b}_i - \Delta\dot{L} + \Delta\dot{B} \tag{347}$$

Note that all the above terms are given for the time $t = t_m$

The next consideration is the possible differentiation between the link velocity error and the velocity lag error due to functional effects.

Let us first examine the link transmission error effects. From basic theory it is known that the phase length path due to the ionosphere is reduced by an amount identical to the group path energy delay.

The phase length decrease relative to free space is given as:

$$\Delta\phi = \frac{-K}{f^2} \int^s N(s)\,ds \cdot \frac{2\pi}{\lambda} \tag{348}$$

If the line integral which defines the total electron content along the LOS is time-varying as it would be due to relative motion of the LOS, the rate of change of the integral will result in a frequency shift of

$$\Delta f = \frac{-K}{f^2} \cdot \frac{2\pi}{\lambda} \cdot \frac{1}{2\pi} \frac{d}{dt} \int^s N(s)\,ds$$
$$= \frac{-K}{cf} \frac{d}{dt} \int^s N(s)\,ds \tag{349}$$

In order to determine the magnitude of the doppler shift due to ionosphere changes, we need to make two basic dynamic considerations:

(1) What is the angular change of the LOS for a constant homogeneous density atmosphere?

(2) What is the time variation for a true horizontal gradient in the vertical electron content?

Consider the evaluation by means of the following:

$$\int N(s)\,ds \doteq I_v \overline{\csc}\,\theta_R \tag{350}$$

where I_v = vertical electron content/m^2

$\overline{CSC}\ \theta_R$ = geometric obliquity factor.

The time derivative of the above integral may be denoted as:

$$\frac{d}{dt}\int N(s)\,ds = \frac{dI_v}{dt}\ \overline{CSC}\ \theta_R + I_v \frac{d}{dt}\ \overline{CSC}\ \theta_R \tag{351}$$

The rate of change of I_v with time can be established by letting the horizontal gradient be given as:

$$\begin{aligned} \frac{dI_v}{dx} &= 1\% \text{ per } 100 \text{ miles} \\ &= 10^{17} \times 0.01 \times 10^{-2} \text{ electrons/m}^2\text{/mile} \\ &= 10^{13} \text{ electrons/m}^2\text{/mile} \end{aligned} \tag{352}$$

Next consider that the maximum relative velocity is given as about 4000 ft/sec so that the rate with respect to time is:

$$\begin{aligned} \frac{dI_v}{dt} &= 10^{13}\ \frac{\text{electrons}}{m^2} \cdot \frac{4000 \text{ ft}}{6000 \text{ ft}} \cdot \frac{1}{\text{sec}} \\ &\cong 0.7 \times 10^{13}\ \frac{\text{electrons}}{m^2}\ \frac{1}{\text{sec}} \end{aligned} \tag{353}$$

The second portion of the rate dynamics is given as:

$$I_v \frac{d}{dt} CSC\ \theta_R = I_v \cot \theta_R\ CSC\ \theta_R\ \frac{d\theta_R}{dt} \tag{354}$$

Geometrically the angular rate for the LOS is defined as:

$$\omega = \frac{d\overline{\theta}_R}{dt} = \frac{v \times R}{|R|^2} \tag{355}$$

Assuming that the range and velocity vectors are orthogonal and that

$$v_{max} \cong 4000 \text{ ft/sec}$$

$$R_{min} \cong 20{,}000 \text{ miles}$$

the angular rate is given as:

$$\omega = \frac{v}{R} = \frac{4000 \text{ ft/sec}}{20000 \times 6000 \text{ ft}} = \frac{1}{5 \times 6000} = \frac{10^{-4}}{3} \frac{\text{rad}}{\text{sec}} \tag{356}$$

The rate change is maximum at elevation angles near the horizon and we can evaluate the function at a low elevation angle of 10 degrees, so:

$$= I_v (5.6)(5.8) \frac{d\theta_R}{dt}$$

$$= I_v (10^{-3})$$

$$= 10^{14} \frac{\text{electrons}}{m^2} \frac{1}{\text{sec}} \tag{357}$$

We may thus establish that both effects (even with an order of magnitude increase in the horizontal gradient) place a bound of about

$$\Delta f = \frac{K}{cf} 10^{14}$$

where $K = 132$ and Δf = cps (Hz)

At "L" band frequency for a generic 621B system, this becomes:

$$\Delta f = \frac{132}{983 \times 10^6 \text{ ft/sec}} \times \frac{10^{14}}{1575 \times 10^6 \text{ Hz}}$$

$$= \frac{132 \times 10^{14}}{1550 \times 10^{15}}$$

$$= 0.85 \times 10^{-2} \text{ Hz} \tag{358}$$

This also converts to

$$\Delta f \cdot \frac{c}{f} = \frac{983}{1575} \times 0.85 \times 10^{-2} \cong 0.5 \times 10^{-2} \text{ ft/sec} \tag{359}$$

The results -- for changes in the angular LOS and for horizontal gradients -- indicate that, at L band, extremely negligible velocity errors will be introduced into the system.

Note also that higher-order ionospheric error terms are also negligible because they depend on reciprocal powers of frequency and with smaller coefficients than the first-order term evaluated above.

It is therefore concluded that: An ionospheric velocity error state is so small as to be unobservable with respect to the finite lag and bandwidth effects normally encountered. The dynamic doppler lag effect is by far the most significant error term.

APPENDIX XIII

PROPAGATION DELAY COMPENSATION

The effect of the earth's atmosphere produces both a curvature in the propagation path and a decrease in propagation velocity along this path as well, so that line-of-sight range measurements are significantly greater than the true geometric range between the emitter and the transmitter. These effects, and compensation algorithms for them, are discussed in this appendix.

1. OVERALL RANGE ERROR COMPENSATION

The range error compensation can be developed by considering the geometry shown in Figure 39. The apparent range which is measured along the curved line-of-sight is given as:

$$Re = \int_{o}^{R} n dR \tag{360}$$

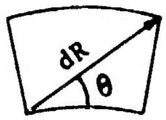
where Re = apparent range

n = atmospheric radio refraction index

R = ray path

The apparent range is a line integral along the ray path which has a non-unity index, since its is generally slightly greater than 1.0. The range increment can be alternately defined in terms of altitude by considering a small spherical shell of height dh. Using differential geometry:

dh



$$\sin\theta = \frac{dh}{dR}$$

$$dR = \frac{dh}{\sin\theta} = dh \text{ CSC } \theta$$

Equation (360) can be written as:

$$Re = \int_{h_1}^{h_2} n(h) \text{ CSC } \theta(h) \, dh \tag{361}$$

where $n(h)$ = functional variation of refractive index with altitude

$\theta(h)$ = variation of elevation angle with altitude.

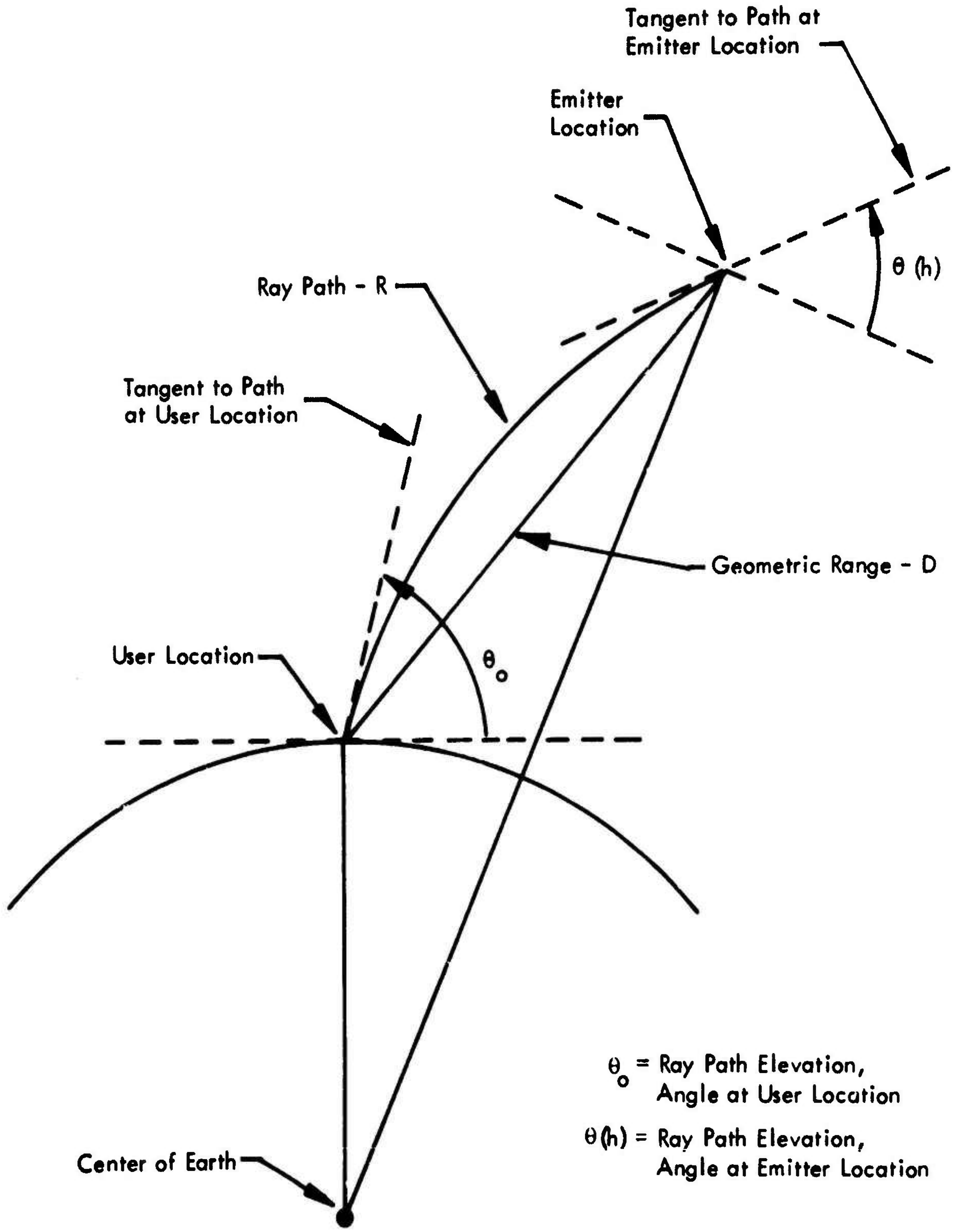


Figure 39. Propagation Geometry

Since the variation of n from unity is very small, another term is defined which represents the small portion which differs from one or:

$$N = (n-1) \times 10^6 \tag{362}$$

where N = refractivity

and $n = 1 + N \times 10^{-6}$

Using equation (362) leads to:

$$Re = \int_{h_1}^{h_2} \csc\theta(h)\, dh + 10^{-6}\int_{h_1}^{h_3} N(h) \csc\theta(h)\, dh \tag{363}$$

Since the geometric range is given as D, the range error may be defined as:

$$\Delta R = Re - D$$

$$\Delta R = \underbrace{\int_{h_1}^{h_2} \csc\theta(h)\, dh - D}_{\substack{\text{curved path diff.} \\ \Delta R_g}} + \underbrace{10^{-6}\int_{h_1}^{h_2} N(h) \csc\theta(h)\, dh}_{\substack{\text{velocity diff.} \\ \Delta R_N}} \tag{364}$$

Equation (364) contains two distinct error terms; the first is the range error due to bending, ΔR_g, and the second is due to velocity decreases, ΔR_N. Generally the bending error, ΔR_g, is an order of magnitude smaller than ΔR_N, and a direct closed-form compensation for it is therefore not contemplated. The bending error is thus a residual error effect term which must be considered in designing the Kalman filter for lumped, residual propagation error estimation.

2. BENDING ERROR COMPENSATION

In this connection, the functional form of the bending error is of interest, and this may be determined by examining empirical data for the error effect. Magnitudes of bending error, ΔR_g, are shown in Figure 40 and in Table LXXXV, as obtained from references 1 and 2.

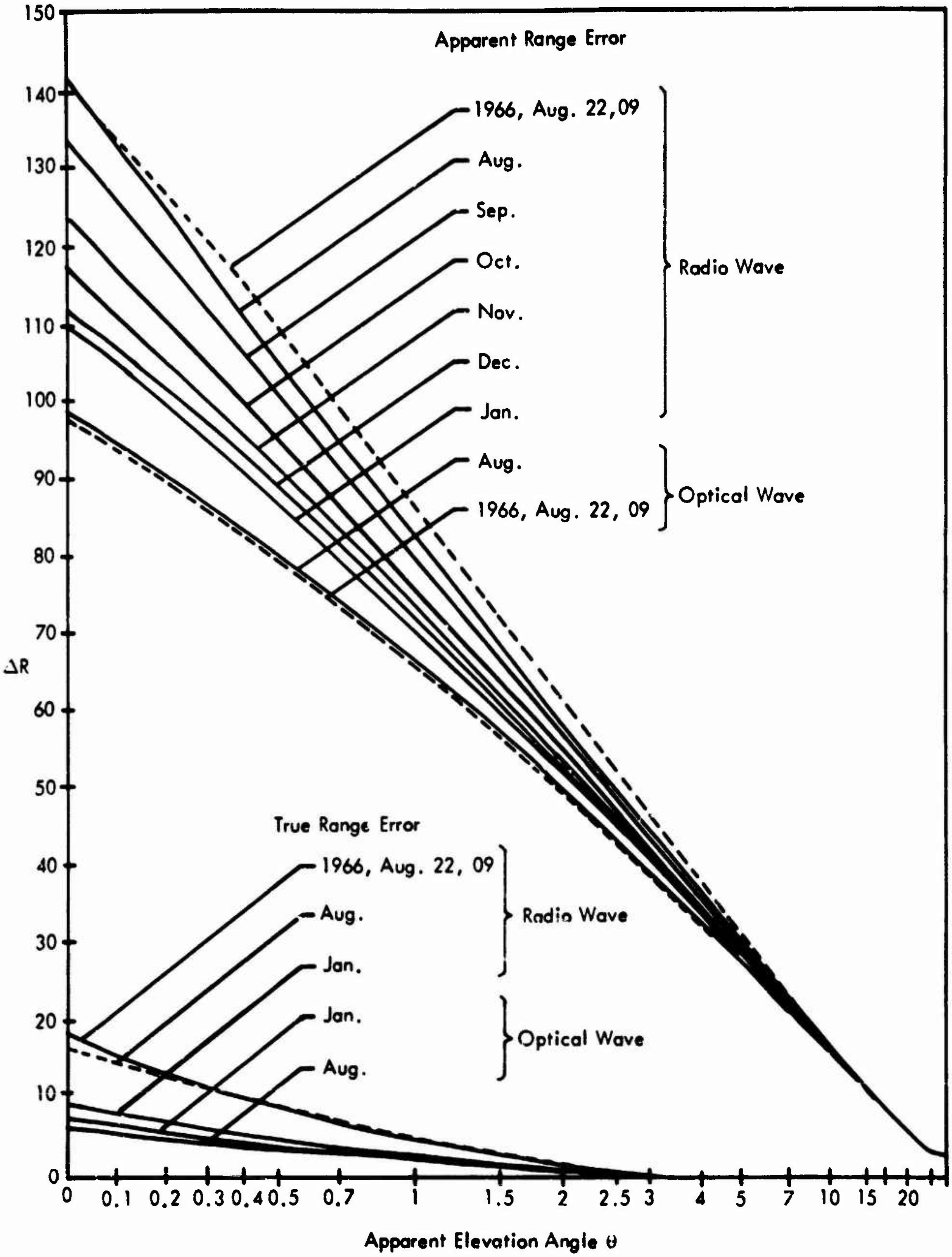


Figure 40. Errors in Apparent and True Range due to the Troposphere

TABLE LXXXV. TYPICAL AND EXTREME VALUES OF RANGE ERRORS FOR TARGETS BEYOND THE ATMOSPHERE

	Typical N_ϵ=320			Extreme N_ϵ=400			Maximum percent
θ_0	ΔR_g	ΔR_N	ΔR_e	ΔR_g	ΔR_N	ΔR_e	$\Delta R_g/\Delta R_e$
	meters						
0	10	100	110	60	165	225	27
20 mrad	2.5	62.5	65	4.5	73	77.5	6
50 mrad	0.7	38.1	38.8	1.0	43	44	2.3
100 mrad	0.14	22.26	22.4	0.2	24.8	25	0.8
200 mrad	0.02	11.9	11.9	0.03	13.0	13.0	0.23
500 mrad	0.001	5.01	5.01	0.002	5.50	5.50	0.04

The information from these two sources is for range to a stationary satellite and the data is replotted in Figure 41, along with a proposed functional curve fit to the data.

The functional curve fit is chosen by assuming the following form:

$$\Delta R_g = K \csc \sqrt{A^2 + \theta_0^2} \tag{365}$$

where K = constant determined by error at zero elevation angle

A = constant which establishes the slope of the error function.

From the empirical data the typical values of the constants are defined as:

$$\left.\begin{aligned} K &= 0.09 \\ A &= 0.3^\circ \end{aligned}\right\} \Delta R_g \text{ in meters}$$

For apparent elevation angles of $\theta_0 \leq 3$ degrees, the bending error is a finite value and should be considered. The only difficulty with using equation (365) is that knowledge of the apparent elevation angle will be in error. From data given by reference 2, the error in geometric and apparent elevation angles is defined as:

$$\epsilon = \theta_0 - \beta \tag{366}$$

where β = geometric LOS angle.

For typical atmospheric conditions the following maximum angular error can be encountered:

$\theta_0 = 0^\circ$ ϵ = maximum of 20 mrad

$\theta_0 = 3^\circ$ ϵ = maximum of 10 mrad

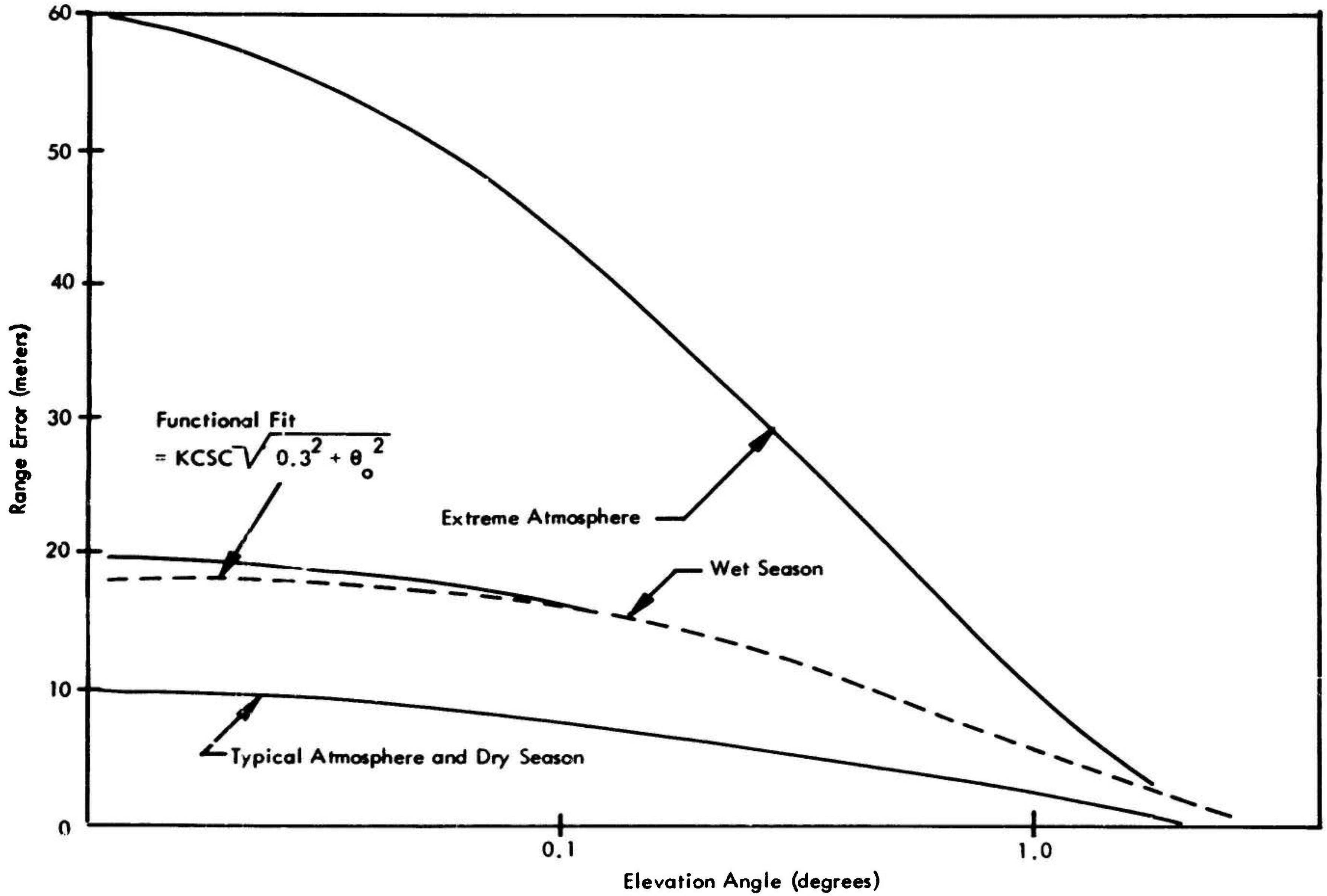


Figure 41. Bending Error Data Fitting

Use of $\theta_0 \cong \beta$ in the residual error equation leads to a negative angular error of 0.5 to 1.0 degree. For this low elevation angle condition, the best option is to initialize the value of equation (365) at its maximum ($\theta_0 = 0°$) and to utilize the pessimistic error state condition.

Note that the angular error, ϵ, is bounded by the total angular bending angle as:

$$\frac{\tau}{2} \le \epsilon \le \tau$$

where τ = angular bending angle, and that expressions exist for ϵ as a function of τ, the surface refraction, refraction profile, and apparent elevation angle.

3. VELOCITY VARIATION COMPENSATION

Returning now to the more important velocity-variation error effect, the principal mechanism involved here is the increase in the energy path length relative to the free space line of sight which is caused by the ionized electrons along the path.

The group path range increase is given by:

$$\Delta R = \int^{S} (u - 1)\, ds \tag{367}$$

where u = group refractive index

S = ray path

The group refractive index can be approximated by the reciprocal of the <u>true</u> index of refraction, so

$$u \simeq \frac{1}{n}$$

$$\therefore (u - 1) = \left(\frac{1}{n} - 1\right) = \left(\frac{1 - n}{n}\right) \simeq (1 - n) \quad \text{since} \quad n \simeq 1.$$

$$\Delta R = \int^{S} (1 - n)\, ds = \int^{S} \Delta n \; ds \tag{368}$$

where $\Delta n = \frac{bN}{\omega^2}$

N = number density of free electrons, electrons per meter^3

ω = angular frequency of incident wave

b = constant = $\frac{e^2}{2\epsilon_o m}$ = 1.6×10^3 (MKS)

where $\frac{e}{m}$ = charge to mass ratio of electron $\quad \epsilon_o$ = free space permitivity

With a little conversion work the path length increase can be written as:

$$\Delta R = \frac{K}{f^2} \int^S N(s)\, ds \tag{369}$$

where ΔR = meters

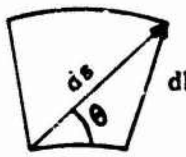
K = 40.3

f = frequency in Hz

$N(s)$ = density function along the ray path given as electrons/meter^3.

Equation (369) shows the basic area of uncertainty in predicting the path error; i.e., that of defining the density function of electrons along the ray path. Alternate approaches to this problem have been considered as follows.

Assume for the moment that the path length error is governed by a density function of electrons which have only a vertical variation. The pertinent elemental path geometry is as shown below.



$\sin\theta = \frac{dh}{ds} \qquad \theta$ = elevation angle

$ds = dh/\sin\theta = dh\ \mathrm{CSC}\,\theta$

and equation (369) becomes:

$$\Delta R = \frac{K}{f^2} \int^H N(h)\ \mathrm{CSC}\,\theta\, dh \tag{370}$$

where H = height between source and user.

Since the ionosphere introduces bending of the ray along the energy path it is really more accurate to express equation (370) as:

$$\Delta R = \frac{K}{f^2} \int^{H} N(h) \text{ CSC } \theta(h) \; dh \tag{371}$$

since CSC θ(h) will be a varying elevation angle as the wave progresses along the ray path due to refraction.

The integral in equation (371) is simplified by defining a mean equivalent value for the function

$$\left\langle \text{CSC } \theta(h) \right\rangle \triangleq \overline{\text{CSC } \theta_R} \qquad \theta_R = \text{reference altitude angle}$$

and

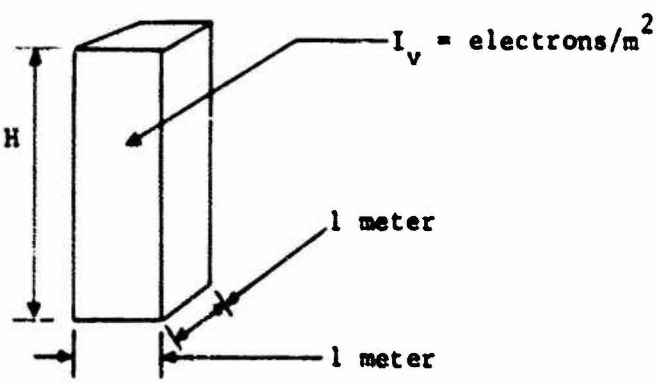
$$\Delta R = \frac{K}{f^2} \; \overline{\text{CSC } \theta_R} \int^{H} N(h)\, dh \tag{372}$$

The term $\int^{H} N(h)\,dh$ is defined as the integrated density profile of a vertical column and in units it is expressed as

$$\text{electrons/meter}^2 = \int^{H} N(h)\, dh$$

and $\qquad I_v = \text{VERTICAL ELECTRON CONTENT} = \int^{H} N(h)\, dh$

The term I_v can be visualized as the number of electrons contained in a vertical column whose cross-sectional area is one square meter:



In some literature the integrated density of the vertical column is given the name:

I_v = TOTAL ELECTRON CONTENT

By way of summarizing it is possible to state

$$I_v = \int^H N(h)\,dh$$

$$= \frac{\int^H N(h)\ \mathrm{CSC}\ \theta(h)\,dh}{\overline{\mathrm{CSC}\ \theta}_R} \tag{373}$$

Note that $I_v \cdot \overline{\mathrm{CSC}\ \theta}_R = I_s$ = total electron content in slant-range column.

The utility of defining I_v = total electron content is that soundings of the ionosphere at reference locations can easily establish the empirical value of the vertical electron content. Or I_v can be obtained as a physical measurement using dual frequency, faraday rotation or ionosonde signals. Hence much information and data exists to define $\hat{I}_v$ about the earth's surface.

Using the concept of a mean equivalent equation for the CSC θ_R, we can write equation (371) as:

$$\Delta R = \frac{K}{f^2}\, I_v\, \overline{\mathrm{CSC}\ \theta}_R \tag{374}$$

where $\hat{I}_v$ = estimated or measured total electron content

$\overline{\mathrm{CSC}\ \theta}_R$ = reference value of mean equivalent

and we know

$$\hat{\overline{\mathrm{CSC}\ \theta}}_R = \frac{\int^H N(h)\ \mathrm{CSC}\ \theta(h)\,dh}{\hat{I}_v} \tag{375}$$

Examination of equation (375) points out the problem of utilizing equation (374) in establishing the mean equivalent CSC θ_R by knowing the <u>entire</u> profile and path length models -- which are very difficult to measure in real time. The solution to the problem is to:

(a) Define a model for N(h) as a function of altitude

(b) Execute a ray tracing program and determine the value of $\overline{\csc\theta_R}$ vs θ.

Two reference sources[3,4] have carried out the above steps and the data are presented in Figure 39 as a function of the elevation angle θ_o which defines the LOS between the emitter and user. This function has been defined alternatively as either (a) the obliquity factor, or (b) the ray-path length adjustment.

Several numerical approximations exist for calculating the "obliquity factor;" two of these are:

$$\overline{\csc\theta_R} = \csc\left(\theta_o^2 + 10^2\right)^{1/2} \qquad \text{[Ref 5]}$$

$$= \csc\left(\theta_o^2 + 20.3^2\right)^{1/2} \qquad \text{[Ref 6]}$$

For comparison these approximations were calculated (Table LXXXVI) and the latter function,

$$\csc\sqrt{\theta^2 + 20.3^2}$$

θ in degrees

is plotted in Figure 42 to show its substantive agreement with the Hughes ray tracing data. The TRW approximation[5] gives a very large adjustment factor at low angles and is rejected for this reason as not agreeing with available data.

The suggested calculation for the ionospheric delay compensation term is thus given as

$$\Delta R = \frac{K}{f^2}\, \hat{I}_v \csc\sqrt{\theta_o^2 + 20.3^2} \tag{376}$$

where $K = 1.31 \times 10^{-16}$

θ_o = LOS elevation angle, degrees

$\hat{I}_v$ = estimate of total electron content in electrons/meter^2

f = carrier frequency, GHz

and ΔR = feet.

TABLE LXXXVI CALCULATION OF $\overline{\text{CSC } \theta_R}$

θ	$\sqrt{\theta^2+10^2}$	Sin( )	CSC( )
(a) TRW Model			
0	10°	0.1736	5.8
10	14.4	0.25	4.0
20	22.6	0.38	2.6
30	31.3	0.52	1.4
40	41.0	0.656	1.5
50	51.0	0.777	1.3
60	61.0	0.874	1.15
70	70.8	0.944	1.06
80	80.6	0.987	1.01
90	90.5	1.0	1.0
(b) Aerospace Model			
0	20.3	0.347	2.9
10	22.6	0.384	2.6
20	28.5	0.477	2.1
30	36.2	0.59	1.7
40	45.0	0.707	1.42
50	54.0	0.809	1.24
60	64.0	0.895	1.12
70	73.0	0.956	1.04
80	82.5	0.99	1.0
90	90.0	1.0	1.0

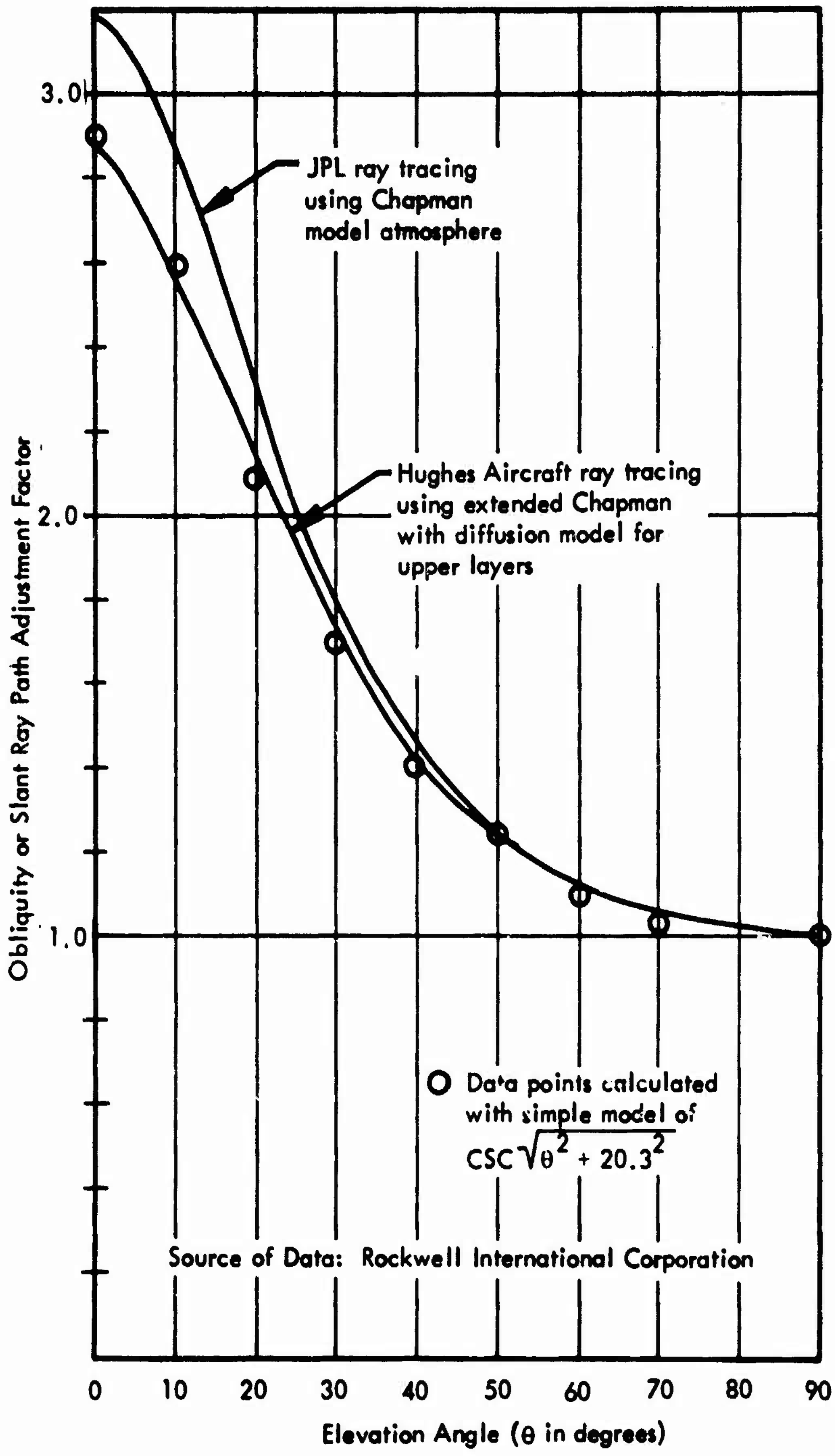


Figure 42. Graphical Function for $\overline{CSC}\theta_R$

Note that this approach does not employ any consideration of a lateral gradient in the electron density but that this limitation can be easily overcome when better ray tracing information is provided simply by altering the constant term under the square root sign.

The calculation of the total electron content depends upon a form of deterministic behavior which is a function of:

(a) Time of day

(b) Time of the year

(c) Phase of the solar cycle

(d) Geographical latitude of signal traversal.

From a software concept for aircraft navigation, the compensations for time of day and for latitude are of primary interest for real-time calcuiations. Simple seasonal and solar cycle compensations can also be incorporated within the software for additional flexibility.

The average value[7] of total electron content for a vertical column is:

$$I_v = 10^{17} \text{ electrons/meter}^2$$

The variation about this average value can be at least as large as an order of magnitude in either direction due to the effects of time and location variability.

A portion of I_v can be compensated for the daily time variation or diurnal effect as a function of time, or more precisely as a function of the location of the sun. The first form of time compensation considered is defined as:

$$f_T = A + \frac{B-A}{12} t \qquad (0 \le t \le 12 \text{ hours})$$

$$= B + \frac{(A-B)}{12}(t-12) \qquad (12 \le t \le 24 \text{ hours})$$

This is a linear function of time which is monotonic up to high noon from a night-time low level of A. The peak level B is reached at high noon. For data plots considered, A is about 4×10^{16} and B is about 3×10^{17} with deviations around these values of about a factor of 3. The difficulty with this linear triangular function is that its symmetry around noon does not comply with the natural phenomena.

A more appropriate function would be one that is asymmetric with a low value at sunrise and a peak in the late afternoon. The linear gradient of the late afternoon decrease is also greater than that of the early morning growth from the night-time value.

The restriction to a linear functional expression with time should be reexamined, and in fact has been in several new investigations which have formulated the following nonlinear functions for diurnal variation.

The suggested method proposed by a Stanford Electronics Lab study[8] is to establish a Fourier series representation to describe the total electron content or

$$I_v(t) = a_o + \sum_{k=1}^{n} \left[a_k \cos kT + a_{kth} \sin kT \right]$$

where $T = \frac{\pi t}{12}$

The coefficients for such a series representation are themselves functions of the day of the year, d, and the solar activity index, s. These functions have the form of power series expansions modifying a Fourier series with yearly harmonics.

The generation of about 13 coefficients was found to be sufficient for describing a_n each day at a specific reference location where data is being observed and collected. Since such locations are limited and unique, the problem of defining the total electron content some distance away from the reference location, but still within its region of correlated occurrence , is accomplished by defining an expansion with a so-called domain.

REFERENCES:

1. Takahashi, Kozo, "Atmospheric Error in Range and Range-Rate Measurements Between a Ground Station and an Artificial Satellite," IEEE Trans. Aerospace and Electronic Systems, Vol AES-6, No. 6, November 1970.

2. Bean, B. R., and E. J. Dutton, "Radio Meteorology," National Bureau of Standards Monograph 92, March 1966.

3. System 621B, Satellite System for Precise Navigation, Vol. II, AD395536, Hughes Aircraft Co. (Secret) 1969.

4. Tracking System Analytic Calibration Activities for the Mariner Mars 1969 Mission, JPL TR 32-1499, 1970.

5. Integrated Inertial/Nav Sat Study Analysis and Planning for Test, Vol. I, AFAL TR-71-319, TRW, 1971 (Confidential).

6. Woodford and Dutcher, "A Satellite System to Support an Advanced Air Traffic Control Concept", Proc. IEEE, March 1970.

7. Lawrence, R., and C. Little, "A Survey of Ionospheric Effects upon Earth-Space Radio Propagation", Proceedings IEEE, January 1964.

8. Waldman, H., and Ada Rosa, "Prognostication of Ionospheric Electron Content", SAMSO TR-71-82, SEL 71-046, August 1971.

9. Freeman, J. J., "Real-Time Compensation for Tropospheric Radio Refractive Effects on Range Measurements," NASA CR-109, October 1964

10. Yionoulis, S. M., "Algorithm to Compute Tropospheric Refraction Effects on Range Measurements," Journal of Geophysical Research, Vol. 75, No. 36, December 1970.

11. Hopfield, H. S., "Tropospheric Effect on Electromagnetically Measured Range: Prediction from Surface Weather Data," Radio Science, Vol. 6, No. 3, March 1971.

12. Ondrasik, V. J., "The Troposphere," part of JPL Technical Report 32-1499, Tracking System Analytic Calibration Activities for the Mariner Mars 1969 Mission, November 1970.

13. Rodrick, T. L., and T. I. Fine, "Computers for Satellite-Based Navigation and Guidance Systems," Extracted from Satellite-Aided Communications, Navigation and Traffic Control, Purdue University Summer Course, 1972.

14. Cooper, G., "Theory of Position and Velocity Determination," Extracted from Satellite-Aided Communications, Navigation and Traffic Control, Purdue University Summer Course, 1972.

APPENDIX XIV

RESIDUAL PROPAGATION ERROR MODELING

This appendix discusses the Kalman filter modeling of propagation errors which remain after closed-form error compensation.

1. SPATIAL MODELING

The combined effect of residual propagation delays caused by the troposphere, ionosphere, and multipath effects may be described by an error term of:

$$\delta\phi = \text{propagation delay}$$

At any given location in the spatial domain of interest, this propagation delay is characterized as a random variable and is described in statistical terms, one of which could be the mean square value of the delay error or:

$$E\,[\delta\phi(x)\delta\phi(x)] = \sigma^2$$

This mean square term is defined over the entire extent of the spatial variable x and it is also possible to state that at any location that is specifically defined, such as $x = x_o$,

$$E\,[\delta\phi(x_o)\delta\phi(x_o)] = \sigma^2$$

We know from observation of the propagation delay due to variability of the dynamics governing the troposphere, ionosphere, and multipath that the delay error is variable in the spatial dimension, x, and that

$$\delta\phi(x) \neq \delta\phi(x_o)$$

The statistical manner in which we describe this variability is by defining the expected value of the product:

$$E\,[\delta\phi(x)\delta\phi(x_o)] = R(\Delta x) \qquad (378)$$

where $\Delta x = x - x_o$ and R( ) = autocorrelation function.

The biggest difficulty with the approach used to this point is that we do <u>not</u> have any information available to describe the analytical or empirical

form of the autocorrelation function, since global autocorrelation data has not been developed to date. In the face of such ignorance, let us stipulate that possibly nature has done the following:

$$R(\Delta x) \cong \sigma^2 \left[K_1^{\,2} + K_2^{\,2} \exp\left(-|\Delta x| \cdot d^{-1}\right)\right] \tag{379}$$

where $K_1^{\,2}\sigma^2$ = fixed bias variance

$K_2^{\,2}\sigma^2$ = distance variable variance

d = correlation distance in feet.

Note that we scale the constants such that

$$\sum_{n=1}^{2} K_n^{\,2} = 1$$

The assumed autocorrelation function states two main ideas:

(a) A fixed portion of propagation delay is to be expected given by the variance term $K_1^{\,2}\sigma^2$.

(b) A portion of the propagation delay decorrelates as an exponential function of distance, d, away from some specific location. (This assumption is not backed up by any experimental data but evaluation of random Omega propagation variations indicates that such an isotropic diffusion does seem to occur for the mean square value of delay.)

The approach to be followed, given that $R(\Delta x)$ is known, will now be developed and can utilize either the assumed function shown in equation (379) or simply leave $R(\Delta x)$ to be defined as a general autocorrelation function; its specific form is not critical to the ensuing development.

Incidentally, some mention should be made at this point of the effect on the delay error of the geometric orientation of the line of sight. This variability does in fact determine the mean square value of the delay and it could be stipulated that $R(\Delta x)$ is really functional in the line-of-sight angle, denoted as θ, so that a spatial autocorrelation is really denoted as:

$$R(\Delta x), \theta) = \text{multivariable function}$$

For a fixed geometric orientation, or $\theta = \theta_o$, the multi-varied function reduces to single dimensionality and so one should think that the further discussions are valid for a specific geometric orientation which is always employed regardless of specific spatial location.

Since several speculative assumptions have already been made, it seems reasonable to mention here that the geometric variability is probably most evident in the constant K_1 and is probably of the form

$$R(\Delta x,\theta) = \sigma^2\left[K_1(\theta)^2 + K_2^2 \exp\left(-|\Delta x|\cdot d^{-1}\right)\right]$$

and $$K_1(\theta) = K_o \csc\theta$$

Some dependence of K_2 on θ also may exist, but is probably negligible.

2. EFFECT OF MODEL CORRECTIONS

The basis for every known form of propagation delay correction is to utilize a model of the delay effect and to apply the model term as a correction. Generally the model is said to be an exact evaluation of $\delta\phi(x_o)$ at the location x_o, or at best a very large part of it, so that we can now formulate the residual error at any location as:

$$\Delta\phi(x) = \delta\phi(x) - \hat{\delta\phi}(x_o) \tag{380}$$

where $$\hat{\delta\phi}(x_o) = \delta\phi(x_o) + n\delta\phi(x_o)$$

or $$\Delta\phi(x) = \delta\phi(x) - \delta\phi(x_o) - n\delta\phi(x_o) \tag{381}$$

where $$n = \text{fractional part}$$

The use of an error term in the correction which is a fractional part of the original correction itself seems to be justified, since generally the models are exact in terms of functional relations but are in error by knowledge of a multiplicative constant of the function.

Next to be determined is the autocorrelation for the residual error in the propagation delay due to the correction; i.e., the expectation of:

$$E\,[\Delta\phi(x)\Delta\phi(x)] \triangleq R_\Delta(\Delta x) = \text{residual autocorrelation}$$

Substituting produces:

$$R_\Delta(\Delta x) = E\left[\left(\delta\phi(x) - \delta\phi(x_o) - n\delta\phi(x_o)\right)^2\right]$$
$$= E\left[\left(\delta\phi(x) - (n+1)\delta\phi(x_o)\right)^2\right]$$
$$= E\left[\delta\phi(x)^2 - 2(n+1)\delta\phi(x)\delta\phi(x_o) + (n+1)^2\delta\phi(x_o)^2\right]$$
$$= E\left[\delta\phi(x)^2\right] - 2(n+1)E\left[\delta\phi(x)\delta\phi(x_o)\right] + (n^2+2n+1)\,E\left[\delta\phi(x_o)^2\right] \qquad (382)$$

Using (377) and (378), we can write:

$$R_\Delta(\Delta x) = \sigma^2 - 2(n+1)\sigma^2\left[K_1^2 + K_2^2 \exp\left(-|\Delta x|\,d^{-1}\right)\right] + (n^2+2n+1)\sigma^2$$
$$= \sigma^2\left(2+2n+n^2 - (2n+2)\left[K_1^2+K_2^2 \exp\left(-|\Delta x|\,d^{-1}\right)\right]\right) \qquad (383)$$

The results of equation (383) have several interesting aspects. Using (383), assume that $n = 0$, which means that an exact correction model is employed for the specific point, x_o; then the residual autocorrelation function is:

$$R_\Delta(\Delta x) = \sigma^2\left\{2-2\left[K_1^2+K_2^2 \exp\left(-|\Delta x|\cdot d^{-1}\right)\right]\right\}$$
$$= \left[\left(2\sigma^2 - 2K_1^2\sigma^2\right) - 2K_2\sigma^2 \exp\left(-|\Delta x|\cdot d^{-1}\right)\right] \qquad (384)$$

The above result states that the residual has a bias-like variance or mean square error of

$$2\sigma^2 - 2K_1^2\sigma^2 = 2\sigma^2\left(K_2^2\right)$$

or

$$R_\Delta(\Delta x) = 2\sigma^2\left(K_2^2\right)\left[1 - \exp\left(-|\Delta x|\,d^{-1}\right)\right] \qquad (385)$$

The result given in (385) can be interpreted as a general result for any given form of correlation process that may occur other than exponential; only the form of the subtracted portion is altered, so that in general for n=0 we have:

$$R_\Delta(\Delta x) = 2\sigma^2\left(K_2^{\,2}\right)\left[1 - R_n(\Delta x)\right]$$

where $R_n(\Delta x)$ = general normalized autocorrelation function.

Again, reviewing equation (383), consider that if the distance variable correlation process does not exist but that $\underline{n \neq 0,}$ the result will then be

$$R_\Delta(\Delta x) = \sigma^2\left[2+2n+n^2 - (2n+2)K_1^{\,2}\right] \tag{386}$$

but $K_1^{\,2} = 1$, since $K_2^{\,2} = 0$

so $R_\Delta(\Delta x) = \sigma^2 n^2$

Hence the bias-like portion of the residual error simply scales as a fraction of the original mean square variance by which the correction estimate is in error.

The spatial autocorrelation model given in equation (383) thus satisfies several specific end-point conditions and also is general enough to accommodate more definitive model alterations as data becomes available. Note that the result of (383) is quite different from the development given in the previous Multilateration report (AFAL TR-72-80). The only significant alteration which may have to be applied to the development is to employ a spatially oriented diffusion process in two spatial dimensions rather than the isotropic one with radial distance utilized in this analysis.

3. TEMPORAL MODELING

The time variations of the residual propagation delays caused by the combined effect of the troposphere, ionosphere, and multipath probably consist of several time-shift-dependent functions which define the time autocorrelation process as:

$$E[\delta\phi(t)\delta\phi(t+\tau)] = R(\tau) \tag{387}$$

and

$$R(\tau) \cong \sigma^2\left[C_1^{\,2} + C_2^{\,2}\exp(-|\tau|\beta) + \sum_{j=3}^{m} C_j^{\,2}\cos(j-2)\,\omega\tau\right]$$

where

ω = angular rotation rate of earth

τ = correlation time variable

β = reciprocal time constant

The basic periodicity implied by the above is a 24-hour variation due to the diurnal changes caused by the sun, with the higher order harmonics being due to improper functional modeling of the daily changes which occur in the troposphere and ionosphere. Note again that

$$\sum_{j=1}^{m} C_j^{\,2} = 1$$

4. COMBINED STATISTICAL ALGORITHM

With both the spatial and temporal propagation errors defined, it is possible to combine both of the autocorrelation functions to describe the complete error statistics. The assumption that spatial errors are not correlated with the temporal errors translates mathematically into:

$$R_\Delta(\Delta x,\tau) = R_\Delta(\Delta x)\cdot R_\Delta(\tau) \tag{388}$$

Using the results of (383) and (387), it is possible to write

$$R_\Delta(\Delta x,\tau) = \sigma^2\left\{(2+2n+n^2)-(2n+2)\left[K_1^{\,2}+K_2^{\,2}\exp(-|\Delta x|\cdot d^{-1})\right]\right\}\cdot\left[C_1^{\,2}+C_2^{\,2}\exp(-|\tau|\beta_c)+\sum_{n=3}^{m} C_n^{\,2}\cos(n-2)\,\omega\tau\right]$$

This correlation process can be broken up and rewritten in several terms, with the distance correlation converted to a time correlation by assuming a constant velocity of movement.

DISTRIBUTION LIST

Activity	Number of Copies
AFAL/TSR WPAFB, Ohio 45433	1
2750th ABW/SSL WPAFB, Ohio 45433	1
SAMSO/XRZE P.O. Box 92960 Worldway Postal Center Los Angeles, California 90009	1
Air University Library Attn: AUL 3T Maxwell AFB, Alabama 36112	1
HQ USAF/SAMID Washington, D.C. 20330	1
AFAL/NVN-1 WPAFB, Ohio 45433	2
AFAL/AAA (J.A. Biernacki) WPAFB, Ohio 45433	26
Defense Documentation Center (DDC) Attn: TIPDR Cameron Station Alexandria, Va. 22314	2
SAMSO/XR Los Angeles AFS Los Angeles, Ca. 90045	1
ESD/XRT L.G. Hanscom Field Bedford, Ma. 01730	1
USAF Academy (DFSFR) Boulder, Co. 80840	1

DISTRIBUTION LIST (Cont'd)

Activity	Number of Copies
AFSC/XRT Andrews AFB 20331	1
ESD/XRC L.G. Hanscom Field Bedford, Ma. 01730	1
HQ USAF/AFRPQPS Attn: W.V. Becker Washington, D.C. 20330	1
AFSC/DLCEC Attn: Dave Anderson Andrews AFB 20331	1

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Northrop Corporation, Electronics Division 1 Research Park Palos Verdes Peninsula, California 90274	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

MULTILATERATION SOFTWARE DEVELOPMENT (PHASE II)

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Technical Report (May 1972 to June 1973)

5. AUTHOR(S) (First name, middle initial, last name)

John D. Weinberg, Myles Harris, E. J. Knobbe, E. R. Kopitzke and Gideon Kochmann (Northrop Corporation) and E. Martin, V. Calbi and M. Bittner (Magnavox Research Laboratories).

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO OF REFS
September 1973	437	14

8a. CONTRACT OR GRANT NO. F33615-72-C-1607 b. PROJECT NO. 6095 02 04 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) NORT 73-270 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFAL-TR-73-297 " 8 NOV 1973

10. DISTRIBUTION STATEMENT

TEST + EVALUATION

Distribution limited to Government agencies only; other requests for this document must be referred to the Air Force Avionics Laboratory (AFAL/AAA), Wright-Patterson Air Force Base, Ohio 45433.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Air Force Avionics Laboratory Air Force Systems Command Wright-Patterson AFB, Ohio 45433

13. ABSTRACT

Historically, vehicle-borne, radio-hybrid navigation system software has too often been designed around preselected navigation hardware on an ad hoc, system-by-system basis. In these developments, little attention has been paid to the inherent physical and functional commonality which underlies much of this superficially quite different software. This report documents the methods, and the very promising results, of the second phase of a software development effort directed at identifying and specifying a standardized, modular, flexible, radio-hybrid navigation system software processor.

The machine-and-language-independent (MLI) processor specification which has in particular been developed to date has been designed so that -- with appropriate, minor, system-specific tailoring -- it can serve as the basic specification for the navigation software development for any specific system, within a wide range of navigation hardware equipment configurations and mission requirements. These currently include any combination of radio (LOS or earth mode), inertial, AHRS, and CADS navigation equipments, as well as the requirements associated with most military and civil aircraft missions and usages. In addition, the MLI processor has been carefully structured to allow for easy accommodation of additional navigation hardware processing requirements. (Continued)

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

ABSTRACT OF AFAL-TR-73-297 (Continued)

The second-phase effort used as its point of departure and developmental framework the basic guidelines, concepts and algorithms established in the initial phase. These include, in particular, the exclusive use of vector-matrix algorithm formulations, processor organization into basic, building-block function- and hardware-specific modules and submodules, the use of a single, mission-phase switchable, computational reference frame, and the use of partitioned, modularly organized Kalman filtering techniques. The overall second-phase effort itself consisted of two main, more or less sequential developments: (a) the extension and refinement of the MLI processor capabilities beyond its first-phase level, and (b) the initial development of a specialized, higher-order language navigation program using the MLI processor specification as a basis.

The improvements of the MLI processor accomplished in the second phase included (a) extension of its navigation hardware applicability to allow use of cheaper AHRU/CADS equipment, either in lieu of or as a backup to an IMU; (b) further development and refinement of a novel and promising radio-autonomous navigation technique; (c) extension and refinement of processor and navigation equipment initialization and alignment techniques; (d) development of a completely partitioned and modularized Kalman filter; and (e) development of a complete set of processor mode control and switching logic specifications. In particular, one of the initialization algorithms developed is a new and powerful one which allows undegraded Kalman filter use of radio pseudoranging measurements, despite large LOS directional uncertainties. Further, the Kalman filter partitioned modularity was achieved without artificial (and performance-degrading) decoupling of interpartition system error dynamics.

Time and money constraints permitted development of the specialized FORTRAN IV/IBM 370 processor program only to a very limited stage. Specifically, all the principal navigation modules required for a single assumed LOS/inertial navigation hardware configuration and navigation mode of operation have been programmed and checked out (for fixed inputs only); no mode switching or control modules have been programmed. However, even this limited level of development was intended (and has served)to accomplish two purposes. First, it provided a learn-by-doing vehicle for the broadly experienced programmer assigned the task, to assay the viability of the MLI processor specifications from the standpoint of real-time programming in either an HOL or a machine-specific language. The preliminary conclusions reached in this regard are that the MLI specification provides the programmer with an extremely flexible and easily modifiable, but standardized approach to programming real-time, multisensor, Kalman (or non-Kalman) navigation system software for any airborne digital computer. In addition, it places overall program efficiency and balance (with regard to execution time and memory requirements) much more completely under the control of the programmer than traditional types of specification.

ABSTRACT OF AFAL-TR-73-297 (Continued)

The second purpose accomplished lies in the fact that the programmed modules thus far developed constitute a nucleus-in-being for further development of either a processor evaluation simulation program on the one hand, or a standardized, real-time HOL master navigation program for subsequent machine-specific translation, on the other.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Algorithm Attitude and Heading Reference System (AHRS) Building Block Central Air Data System (CADS) Higher-Order Language (HOL) Inertial Measurement Unit (IMU) Inertial Navigation Initialization Kalman Filter Line of Sight (LOS) Machine-and-Language-Independent (MLI) Modular Module Multilateration Multisensor Navigation Partitioned Processor Radio-Autonomous Navigation Radio-Hybrid Navigation Radio Pseudoranging Simulation Software Vector-Matrix						